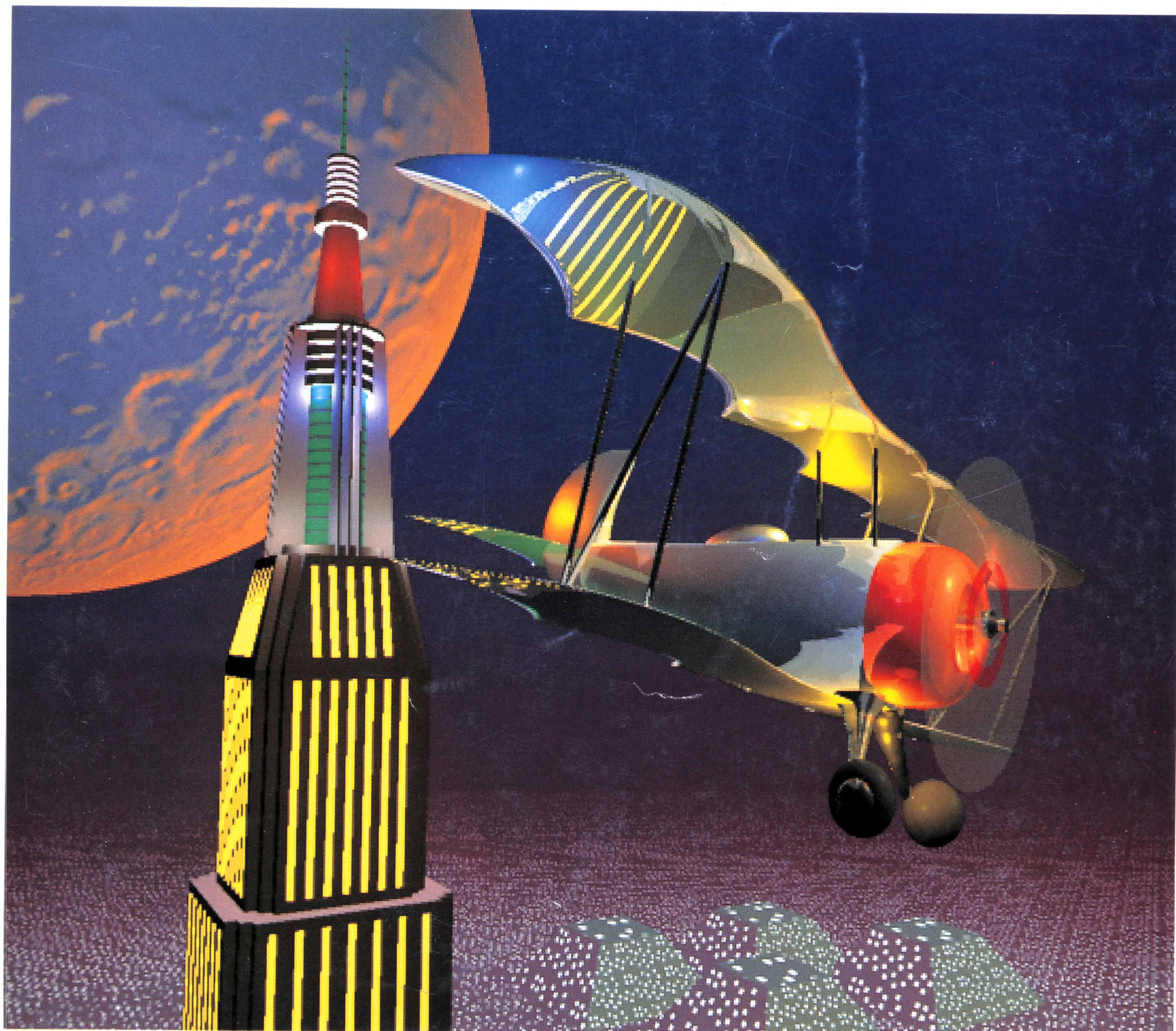
1990年８月１日発行（毎月１回１日発行）第９巻８号通巻100号　昭和58年11月２日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

Oh!X

8

SOFT BANK



SHARP New-Life People

ひらかれた知性。





ザ・ワークステーション。80Mバイトハードディスク、SCSIインターフェイスを標準装備。

SUPER HD

本体＋キーボード＋マウス・トラックボール

CZ-623C-TN(チタンブラック) 標準価格498,000円(税別)

アートの系譜。

EXPERT II

本体＋キーボード＋マウス・トラックボール

CZ-603C-BK(ブラック)・-GY(グレー) 標準価格338,000円(税別)/HDタイプ CZ-613C-BK(ブラック) 標準価格448,000円(税別)

ニュースタンダード。

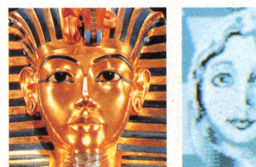
PRO II

本体＋キーボード＋マウス

CZ-653C-BK(ブラック)・-GY(グレー) 標準価格285,000円(税別)

HDタイプ CZ-663C-BK(ブラック)・-GY(グレー) 標準価格395,000円(税別)

資料請求券 X68000 Oh!X 8係



次代のユーザーインターフェイスを象徴する〝SX-WINDOW※〟搭載。

今回のX68000ニューシリーズのデビューに関して、ハードウェア以上にウィンドウ環境の提供に耳目が集中したことは、昨今のビジュアルユーザーインターフェイス事情をふまえれば、当然のことと言えるでしょう。マルチウィンドウを駆使してX68000をコントロールする、待ち望まれていた環境がこのSX-WINDOWによって実現されるのです。何の予備知識もなしにこのウィンドウに接した方は、一見して従来のビジュアルシェルのバージョンアップと思われるかもしれませんが、本質的には全く異質のものと言えます。ひとつのウィンドウである仕事をさせながら、別のウィンドウで違う仕事にとりかかる。ひとことで言えばアプリケーションを実行させる環境としてのウィンドウであるということ。これまでのビジュアルシェルではできなかったシーンを生み出しています。複数のアプリケーションを同じ操作のもとで走らせたり、アプリケーション相互でデータのやりとりが可能になるわけです。そして、次代のインテリジェンスを鮮やかに象徴する4階調のハイセンスな画面処理――。SX-WINDOWをターゲットとしたアプリケーション開発もすでに推進されており、これからの展望という点からも大いに期待されるところです。また、このSX-WINDOWはディスクによって供給され、BIOSの高速化(平均2倍)も含めてOSであるHuman68kの機能を拡張。ニューシリーズのみならず、すべてのX68000でこの新しい環境が享受できます。





※SX-WINDOWの起動には、メインメモリ2MBが必要です。CZ-600C/601C/611C/652C/653C/662C/663CでSX-WINDOWをご使用の際には、あらかじめ別売の1MB増設RAMボードを増設してください。

NEW

X68000
PERSONAL WORKSTATION
SUPER・EXPERT・PRO

充実のディスプレイラインアップ

15型カラーディスプレイテレビ(ドットピッチ0.39mm)	CZ-602D-BK(ブラック)・-GY(グレー※)	標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.39mm)	CZ-605D-BK(ブラック)・-GY(グレー)	標準価格115,000円(スピーカー2個/チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31mm)	CZ-613D-TN(チタンブラック)・-BK(ブラック)・-GY(グレー)	標準価格135,000円(スピーカー2個/チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	CZ-603D-BK(ブラック)・-GY(グレー)	標準価格 84,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	CZ-604D-BK(ブラック)・-GY(グレー)	標準価格 94,800円(スピーカー2個/チルトスタンド同梱・税別)
21型カラーディスプレイ(ドットピッチ0.52mm)	CU-21HD-BK(ブラック)	標準価格148,000円(スピーカー2個同梱・税別)

※印の商品は在庫僅少です。

EXEリーダーズグッズ
プレゼント実施中

●いま、EXE会員よりご紹介のお客様がEXEショップでX68000シリーズを購入されますと、EXE会員にEXEリーダーズグッズをプレゼントします。詳しくはEXEショップにお問い合わせください。
●また、X68000シリーズをご購入のお客様は、ぜひEXEクラブにご入会ください。

●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

シャープ株式会社

Oh!X

CONTENTS

●特集

40 ADVANCED 2D GRAPHICS

44 X68000用グラフィックツール紹介 あなたにあったグラフィックツール　荻窪 圭

50 ギザギザのないグラフィック関数 アンチエリアシングとは?　丹 明彦

68 X-BASICによる画像処理 後処理によるジャギーの除去　中野修一

72 色数の補間と量子化 グラフィックデータを変換する　鈴木康弘

77 4096色→8色変換 Zの画像をX1で　亀田雅彦

●Oh!X通巻100号記念特別企画

23 表紙ぎゃらりぃ

97 対戦ポピュラス　祝一平VS西川善司　浦川博之

100 愛読者特大モニタプレゼント

●シリーズ全機種共通システム

145 THE SENTINEL

146 リンカWLK　石上達也

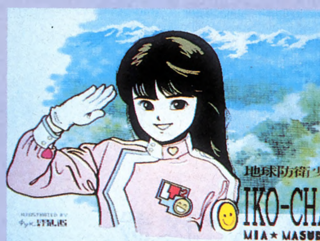
●読みもの

158 第40回　知能機械概論――お茶目な計算機たち―― 人工知能の冒険　有田隆也

160 猫とコンピュータ　第50回 サーチャーでござる　高沢恭子

162 X-OVER NIGHT 第3話 旅行あれこれ　高原秀己

〈スタッフ〉
●編集長／前田　徹　●編集／植木章夫　岡崎栄子　浅井研二　●協力／有田隆也　中森　章　後藤貴行　林　一樹　荻窪　圭　岡本造一郎　毛内俊行　吉田賢司　影山裕昭　相馬英智　古村　聡　村田敏幸　丹　明彦　三沢和彦　長沢淳博　宮島　靖　金子俊一　浦川博之　山田純二　●カメラ／杉山和美　●イラスト／永沢しげる　山田晴久　小栗由香　●アートディレクター／島村勝頼　●レイアウト／元木昌子　AD GREEN　●校正／グループごじら

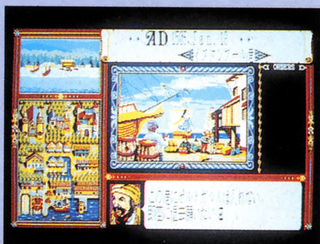


特集　ADVANCED 2D GRAPHICS



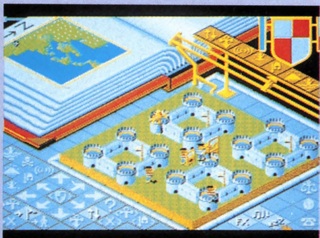
カードゲームHEART

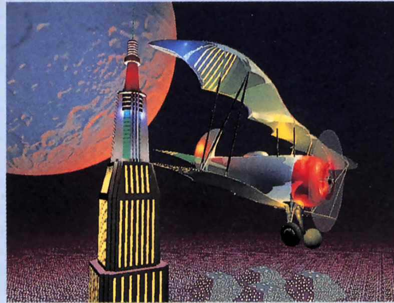

かべくずし


大航海時代



ウルティマⅤ


プロミストランド


表紙絵：須藤　牧人

E N T S

1990 AUG. 8

●THE SOFTOUCH

28	SOFTWARE INFORMATION 話題のソフトウェア	
	GAME REVIEW	
32	大航海時代	浦川博之
34	ウルティマV	荻窪　圭
36	プロミストランド	山田純二
38	AFTER REVIEW 天下統一/ダウンタウン熱血物語 あ〜くしゅ/Yet Another Column	

連載/紹介/講座/プログラム

81	X68000用画像回転プログラム XROT0.X	渡辺伸也
88	X68000 CARD.FNC用カードゲーム HEART・負けるが勝ち	池谷昌彦
92	X1turbo用ディスク管理プログラムINTEGRAL X1 トランジェントコマンドを作る	亀田雅彦
102	PC-E500テーブルトークRPGサポートシステム(1) ポケコンでCARPGを	松井　信
104	ハードウェア工作入門〈2〉 基本インタフェイス回路 その2	三沢和彦
107	X-BASICプログラミング調理実習(13) 超入門・ファイル処理	泉　大介
113	X68000マシン語プログラミングChapter_OFH マウスwithグラフィック	村田敏幸
121	PASCALプログラミングへの招待〈3〉 PASCALのデータ型を見る	藤井義巳・藤木健士
126	マシン語カクテル in Z80's Bar 第14回 楽な逆ポーランド?	山田純二
130	(で)のショートプロぱーてぃ その12 祝！１周年記念	古村　聡
134	Oh!X LIVE in '90 OMENS OF LOVE (X68000) ENDLESS RAIN (X1/turbo) ダートフォックスよりRunning up !(X68000MUSICDRVサンプル曲)	 小玉和博 伏喜義宏 西川善司

ペンギン情報コーナー……164
FILES Oh!X……166
Oh!X質問箱……168
STUDIO X……170
編集室から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssey……174

UNIXはAT&T BELL LABORATORIESのOS名です。
Machはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mplus, CP/M-86, CP/M-68K, CP/M-8000, DR-DOSはDIGITAL RESEARCH
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACRO80, MS CはMICRO SOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事会
WordStar, WordMasterはWORDSTAR International
TURBO PASCAL, TURBO C, SIDEKICKはBOLAND INTERNATIONAL
LSI CはLSI JAPAN
HuBASICはハドソンソフト
の商標です。その他，プログラム名，CPUは一般に各メーカーの登録商標です。本文中では"TM"，"R"マークは明記していません。
本誌に掲載されたプログラムの著作権はプログラム作成者に保留されています。著作権上，PDSと明記されたもの以外，個人で使用するほかの無断複製は禁じられています。

■広告目次

アイツー……186
アイビット電子……190
アクセス……192
アンス・コンサルタンツ……9
エスピーエス……181
AVCフタバ電機……183
オーエーランド……187
OKハウス……182
計測技研……184・185
コナミ……12・13
ザイン・ソフト……11
J&P……表3
システムサコム……14・15
シャープ……表2・表4・1・4-8
ソフトクリエイト……189
九十九電機……22
T&Eソフト……17
デンキヤ……188
パソコンプラザオクト……20・21
P&A……18・19
ビクター音楽産業……16
満開製作所……191(下)
ロゴスシステム……10

SHARP

クリエイティブマインドあふれる周辺機器が



ディスプレイ関連

カラーディスプレイテレビ



15型カラーディスプレイテレビ
CZ-602D-BK
★CZ-602D-GY
標準価格 99,800円（税別）
（チルトスタンド同梱）



15型カラーディスプレイテレビ
CZ-605D-BK・-GY
標準価格115,000円（税別）
（スピーカー2個・チルトスタンド同梱）



15型カラーディスプレイテレビ
CZ-613D-TN・-BK・-GY
標準価格135,000円（税別）
（スピーカー2個・チルトスタンド同梱）

CRTフィルター



高性能CRTフィルター
BF-68PRO
標準価格 19,800円（税別）
（14/15型用）

カラーディスプレイ



14型カラーディスプレイ
CZ-603D-BK・-GY
標準価格 84,800円（税別）
（チルトスタンド同梱）



14型カラーディスプレイ
CZ-604D-BK・-GY
標準価格 94,800円（税別）
（スピーカー2個・チルトスタンド同梱）



21型カラーディスプレイ
CU-21HD
標準価格 148,000円（税別）
（スピーカー2個同梱）

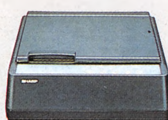
チューナー



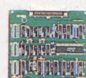
RGBシステムチューナー
CZ-6TU-BK・-GY
標準価格 33,100円（税別）
（リモコン付）

アートツール

画像入力



カラーイメージスキャナ※1
CZ-8NS1
標準価格 188,000円（税別）



スキャナ用パラレルボード
CZ-6BN1
標準価格 29,800円（税別）

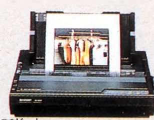
映像入力



カラーイメージユニット※2
CZ-6VT1-BK
CZ-6VT1
標準価格 69,800円（税別）

プリンタ

カラープリンタ



24ドット
熱転写カラー漢字プリンタ
★CZ-8PC3
標準価格 65,800円（税別）
（信号ケーブル同梱）



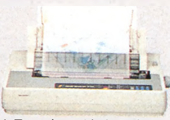
48ドット
熱転写カラー漢字プリンタ
CZ-8PC4
CZ-8PC4-GY
標準価格 99,800円（税別）
（信号ケーブル同梱）

カラービデオプリンタ



カラービデオプリンタ
CZ-6PV1
標準価格 198,000円（税別）
（信号ケーブル同梱）

カラーイメージジェット

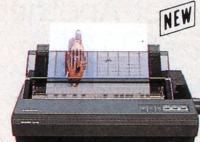


カラーイメージジェット※3
IO-735X
標準価格248,000円（税別）
（信号ケーブル別売）

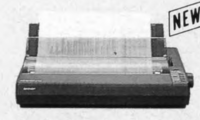
ドットプリンタ



24ピン
カラー漢字プリンタ（80桁）
CZ-8PG1
標準価格 130,000円（税別）
（信号ケーブル同梱）



24ピン
カラー漢字プリンタ（136桁）
CZ-8PG2
標準価格 160,000円（税別）
（信号ケーブル同梱）



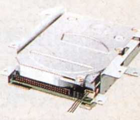
24ピン漢字プリンタ（136桁）
CZ-8PK10
標準価格 97,800円（税別）
（信号ケーブル同梱）

ファイル

ハードディスク



ハードディスクユニット（20MB）
CZ-620H
標準価格 178,000円（税別）



増設用ハードディスク
ドライブ（40MB）
（CZ-602C/603C/652C/653C内蔵用）
CZ-64H
標準価格 120,000円（税別）
（取付費別）
※取付に関してはシャープお客様ご相談窓口にてご相談ください。

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1標準価格29,800円（税別）で接続してください。
※2 CZ-603D/604D、CU-21HDをご使用の場合は、RGBシステムチューナーCZ-6TU（別売）が必要です。
※3 別売の信号ケーブルIO-73CX標準価格5,500円（税別）で接続して下さい。

X1・X1turbo シリーズ用 周辺機器
標準価格は税別です。

カラーディスプレイ		
●21型カラーディスプレイ※1	CU-21HD	148,000円

映像・画像入力編集装置		
●カラーイメージスキャナ	CZ-8NS1	188,000円
●カラーイメージボードII	CZ-8BV2	39,800円

●立体映像セット	★CZ-8BR1	29,800円
●パーソナルテロッパ※2	CZ-8DT2	44,800円

FM音源		
●ステレオタイプFM音源ボード	CZ-8BS1	23,800円

スピーカー（2本1組）標準装備、ミュージックツール同梱

プリンタ		
●24ピンカラー漢字プリンタ（80桁）	CZ-8PG1	130,000円
●24ピンカラー漢字プリンタ（136桁）	CZ-8PG2	160,000円
●24ピン漢字プリンタ（136桁）	CZ-8PK10	97,800円
●24ドット熱転写カラー漢字プリンタ	★CZ-8PC3	65,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4	99,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4-GY	99,800円
●カラービデオプリンタ	CZ-6PV1	198,000円
●カラーイメージジェット	IO-735X	248,000円

ファイル		
●ミニフロッピーディスクユニット（2HD・2D）※3	★CZ-520F	118,000円

シャープ株式会社

●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221（大代表）

資料請求券 X68000 ペリフェラル Oh!X 8月

X68000をサポート。



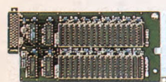
シャープペリフェラルファミリー

X68000



ボード

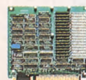
拡張メモリ



1MB増設RAMボード
(CZ-600C専用)
CZ-6BE1
標準価格 35,000円(税別)



1MB増設RAMボード
(CZ-601C/611C/652C/653C/662C/663C用)
CZ-6BE1B
標準価格 28,000円(税別)

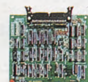


2MB増設RAMボード※4
CZ-6BE2
標準価格 79,800円(税別)

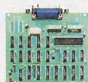


4MB増設RAMボード※4
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス



ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)

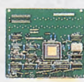


GP-IBボード
CZ-6BG1
標準価格 59,800円(税別)



増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)

数値演算プロセッサ



数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX



FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI



MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

モデム



モデムユニット※5
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル

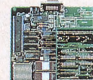


RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)



RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

LANボード



NEW

LANボード
CZ-6BL1
標準価格 268,000円(税別)
CZ-6BL2
標準価格 298,000円(税別)
※電源ユニット・ソフトウェア
(ネットワークドライバVer1.0)同梱

入力



インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)



マウス・トラックボール
CZ-8NM3
標準価格9,800円(税別)



トラックボール
CZ-8NT1
標準価格 13,800円(税別)



マウス
CZ-8NM2A
標準価格 6,800円(税別)



ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他

拡張スロット



拡張 I/Oボックス(4スロット)
(CZ-600C/601C/602C/603C/611C/612C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格 88,000円(税別)

スピーカー



アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック



システムラック
(CZ-600C/601C/602C/603C/611C/612C/613C/623C用)
CZ-6SD1
標準価格 44,800円(税別)

※4 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格35,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。
※5 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

●ミニフロッピーディスクユニット(2D)	★CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)※4	CZ-53F-BK	19,800円

拡張ボード・その他

●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード※5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※6	CZ-8BF1	14,800円

●JIS第1水準漢字ROM※7	CZ-8BK2	19,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張 I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ※8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円

●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド	CZ-6ST1-E・-B	5,800円
●高性能CRTフィルター ※9	BF-68PRO	19,800円
●スキャナ用パラレルボード ※10	CZ-8BN1	27,800円

●品番中の-表示は、B〈ブラック〉・E〈オフィスグレー〉を示します。※1 X1ターボZシリーズ用 ※2 CZ-862Cには接続できません ※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用 ※6 CZ-850CでCZ-520Fを使用する場合に必要 ※7 CZ-800C、801C、802C、803C、811C、820C用 ※8 CZ-820C、822C、830C用 ※9 14/15型用 ※10 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。

★印の商品は在庫僅少です。

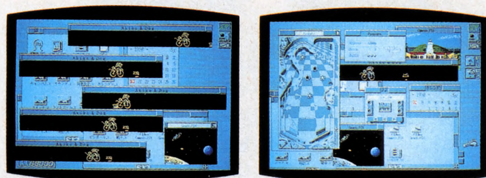
電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

SHARP

"アート"と呼べる高水準のソフトウェアが

次代のインテリジェンス、ウィンドウ環境をあなたのX68000で。

ユーザー本位の操作環境を提供するフル画面マルチウィンドウタイプの美しいデスクトップ(テキスト面/単色4階調＋カラー4色、グラフィック面/カラー65,536色中16色)、イベント・ドリブン型マルチタスク処理により複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。従来のビジュアルシェルとは異なり、今後のアプリケーションソフトが統一された操作環境で実行できるようになります。



SX-WINDOW ver1.0

CZ-259SS 10万台達成ご愛用感謝価格6,800円(税別)



高速通信をサポート。これからの、そしてさまざまな通信環境に対応する高機能コミュニケーションソフト。

Communication PRO-68Kのバージョンアップ版です。300BPSから19,200BPSまでの通信速度に対応し、パソコン同士の接続や各種データベースの漢字端末に、またホストコンピュータとのデータ通信に利用できます。さらにMNPモデムへの対応で、ハードフロー制御(CTS/RTS)をサポート。その他、高速逆スクロール機能、オートログイン/オートパイロットが可能な自動実行機能、コンカレント機能も装備。行入力機能やスクリーンエディタなど豊富な編集機能も魅力です。また、バイナリファイルを転送するプロトコルとしてX modem(128/SUM、128/CRC、1K)、Ymodem(G、BATCH、G-BATCH)、TransIt2(TEXT、BINARY)プロトコルもサポートしています。



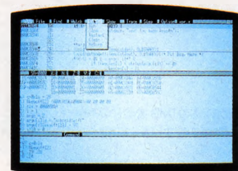
CZ-257CS

標準価格
19,800円(税別)

Communication ver2.0 PRO-68K

ソースコードデバッガをはじめ、各種開発ツールを強化。バージョンアップされたCコンパイラ。

Cのソースレベルでデバッグできるソースコードデバッガを搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman68k ver2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ(約800種)となっています。強力なMAKEも新たに追加。C言語の標準であるANSI規格準拠をさらに強化し、プロトタイプ宣言もデフォルトに変更されました。「BASIC-Cコンバータ」、「アセンブラ」、「リンカ」、「デバッガ」、「ソースコードデバッガ」、「アーカイバ」、「ライブラリアン」、「コンバータ」などのツールが装備されています。



CZ-245LS

7月発売予定

C compiler ver2.0 PRO-68K

資料請求券
CZソフト
Oh!X
8係

シャープ株式会社 ●お問い合わせは…シャープ(株)電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)へ。

X68000をサポート。



シャープオリジナルソフトウェア X68000

ビジネスツール

Hyperword
■CZ-251BS 標準価格39,800円(税別)

X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。



CYBERNOTE PRO-68K
■CZ-243BS 標準価格19,800円(税別)

プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどへの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-200Lが必要)。



Stationery PRO-68K
■CZ-240BS 標準価格14,800円(税別)

他のソフトを起動する前に、このStationery PRO-68Kを一度起動するだけで、他のソフトを実行中にも「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-200Lが必要)。



TOP給与計算エキスパート
■CZ-228BS 標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

CARD PRO-68K
■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

DATA PRO-68K
■CZ-220BS 標準価格58,000円(税別)

入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

TOP財務会計
■CZ-227BS 標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CARD PRO-68K用システム手帳リフィル集
■CZ-241BS 標準価格9,800円(税別)

CARD PRO-68K用活用フォーム集
■CZ-242BS 標準価格9,800円(税別)

BUSINESS PRO-68K
■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。

アートツール

NEW PrintShop PRO-68K
■CZ-221HS 標準価格19,800円(税別)

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビリティツール。ほとんどの処理をアイコンで表示しマウスで選ぶフレンドリーオペレーション。



グラフィックライブラリ VOL.1
■CZ-235GS 標準価格8,800円(税別)

暑中見舞用を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

グラフィックライブラリ VOL.2
■CZ-236GS 標準価格8,800円(税別)

年賀状を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

サウンドツール

Musicstudio PRO-68K ver.1.1
■CZ-252MS 標準価格28,800円(税別)

24トラック対応MIDIマルチレコーディングソフトMusicstudio PRO-68Kがバージョンアップしました。従来の機能に加え、小節間のコピー及びデリートや、MIDIインプットモニターなど、数々の機能を追加・改良。さらに使いやすくなりました。

※MIDIボード(CZ-6BM1)が必要です。



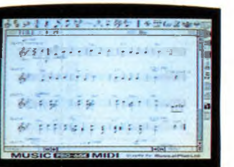
MUSIC PRO-68K〔MIDI〕
■CZ-247MS 標準価格28,800円(税別)

MIDI対応自動伴奏機能をサポート、簡単な楽譜入力でMIDI演奏が楽しめます。

※MIDIボード(CZ-6BM1)が必要です。

ソングライブラリ〈101曲集〉
■CZ-248MS 標準価格8,800円(税別)

鑑賞用と音楽データ加工作成用からなるライブラリです。



Sampling PRO-68K
■CZ-215MS 標準価格17,800円(税別)

AD PCM機能を活かす高機能サンプリングエディタ。多彩なEDITORを装備、サンプリング音のデータはBASICでも活用できます。

SOUND PRO-68K
■CZ-214MS 標準価格15,800円(税別)

スタジオのコンソールパネルを操作する感覚でFM音源による音創りが楽しめるサウンドエディタ。

MUSIC PRO-68K
■CZ-213MS 標準価格18,800円(税別)

最大8パートのスコア(総譜)が書け、内蔵のFM音源で演奏できる楽譜ワープロ&演奏用ツール。

ゲーム



シューティングゲーム
〈ツインビー〉
■CZ-217AS
標準価格7,800円(税別)
©KONAMI 1988



シューティングゲーム
〈沙羅曼蛇〉
■CZ-218AS
標準価格8,800円(税別)
©KONAMI 1989



ブロックゲーム
〈アルカノイド〉
■CZ-222AS
標準価格7,800円(税別)
©TAITO CORP. 1987



ドライブゲーム
〈フルスロットル〉
■CZ-231AS
標準価格8,800円(税別)
©TAITO CORP. 1988



スポーツゲーム
〈熱血高校ドッジボール部〉
■CZ-232AS
標準価格7,800円(税別)
©TECHNOS JAPAN CORP. 1988



アクションゲーム
〈パックマニア〉
■CZ-233AS
標準価格7,800円(税別)
©NAMCO



アクションゲーム
〈ニュージーランドストーリー〉
■CZ-230AS
標準価格8,800円(税別)
©TAITO CORP. 1989



スポーツゲーム
〈V'BALL〉
■CZ-246AS
標準価格7,900円(税別)
©TECHNOS JAPAN CORP. 1989



バイクレーシングゲーム
〈スーパーハングオン〉
■CZ-238AS
標準価格8,800円(税別)
©SEGA 1987



ジェットヘリ・シミュレーションゲーム
〈サンダーブレード〉
■CZ-239AS
標準価格9,500円(税別)
©SEGA 1987



アクションゲーム
〈ダウンタウン熱血物語〉
■CZ-254AS
標準価格8,800円(税別)
©TECHNOS JAPAN CORP. 1989

開発ツール

OS-9/X68000
■CZ-219SS 標準価格29,800円(税別)

OS-9のもつマルチタスク機能、リアルタイム機能を活かした使い易く機能的なOS環境を提供。これまでのデータ資産も活かせます。

※OS-9はマイクロウェア社の登録商標です。

Human68k ver2.0
■CZ-244SS 標準価格9,800円(税別)

THE福袋V2.0
■CZ-224LS 標準価格9,980円(税別)

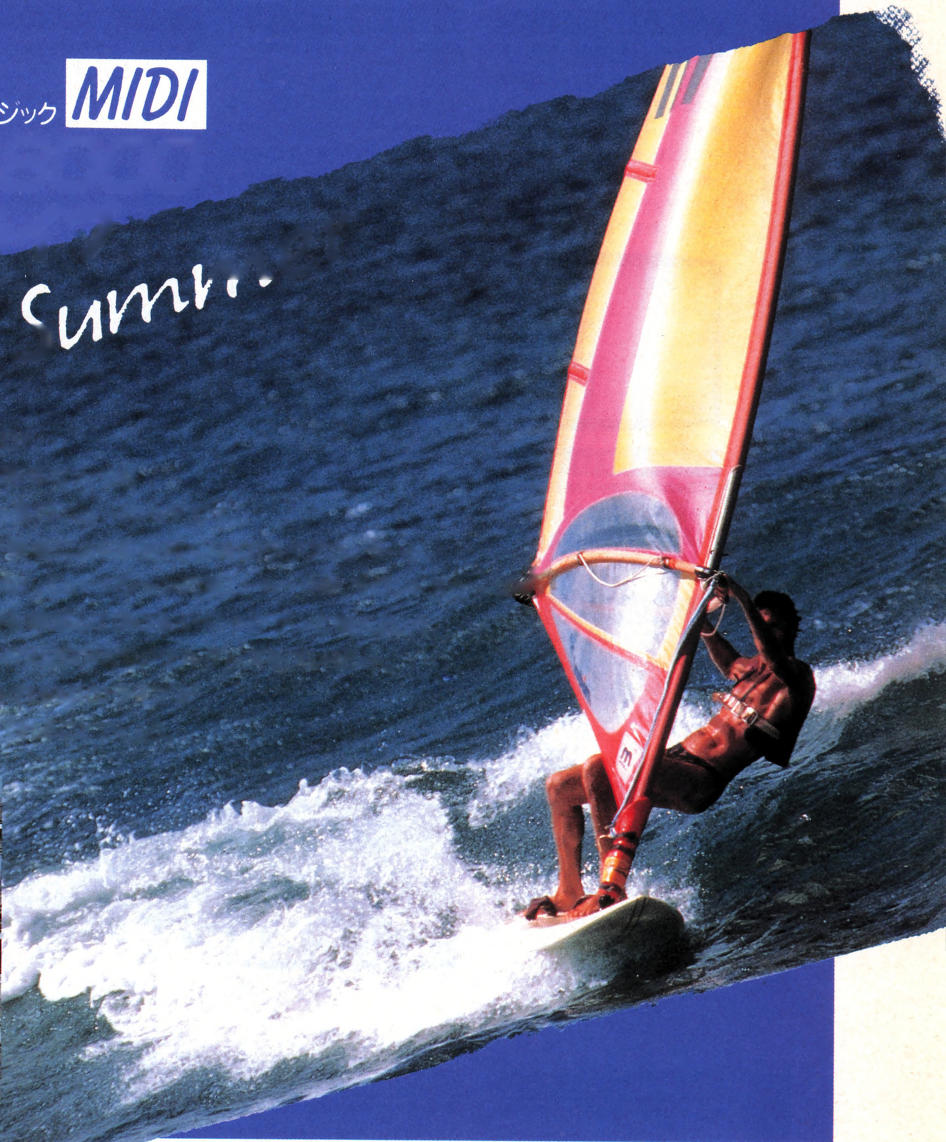
AI-68K(Staff LISP/OPS PRO-68K)
■CZ-234LS 標準価格188,000円(税別)

SHARP

[redacted]mation



パソコンミュージック MIDI





EXEクラブが待っている。

●X68000を手にしたら何はともあれ「EXEクラブ」へ。本体同梱の入会申し込みハガキを送るだけで会員証として、オリジナルデザインのカード電卓がもらえちゃう(会費無料)。EXEクラブニュースや最新ソフト、周辺機器などX68000の最新情報を随時ご案内。各種イベント、フェアへのご招待もあります。

(「X68000は持っているけど、まだ入会してない」方も、ぜひこの機会にお申し込み下さい。)



●EXE会員にはEXEリーダーズグッズ・プレゼントも実施中です。詳しくはお近くのEXEショップまで。

NEW X68000、新作ソフト、面白イベント……
まるごと見・体・験フェア。

●今回のテーマはニューX68000。SUPER-HD/EXPERTII/PROIIの魅力を直にご体験ください。業界注目のSX-WINDOWも必体験。他、新作ソフト体験コーナー、100インチ液晶プロジェクションによる大迫力のゲームたちなど、新しい出会いがあるかもしれません。X68000オリジナルグッズも展示即売。ぜひお近くの会場へお立ち寄りください。

●X68000見体験フェア・音遊サウンドライブ開催日程

開催月日	開催地区	開催場所	お問い合わせTEL	
7/20(金)・21(土)	東京	ソフトクリエイトX68000フェア	03-486-6541	◎
7/22(日)	太田	パソコンランド21太田店X68000フェア	0276-45-0721	◎
7/22(日)	金沢	サンミュージックOAプラザX68000フェア	0762-48-6131	◎
7/22(日)	高松	シャープ見体験フェアイン高松	0878-23-4868	※
7/23(月)	高崎	パソコンランド21高崎東口店X68000フェア	0273-26-5221	◎
7/28(土)・29(日)	札幌	九十九電機札幌店X68000フェア	011-241-2299	◎
7/28(土)・29(日)	富山	シャープ見体験フェアイン富山	0762-49-1181	※◎
7/28(土)・29(日)	神戸	星電社三宮本店X68000フェア	078-391-8171	◎
8/4(土)	高崎	パソコンランド21高崎飯塚店X68000フェア	0273-64-0521	◎
8/4(土)	京都	J&P京都寺町店X68000フェア	075-341-3571	◎
8/5(日)	前橋	パソコンランド21前橋店X68000フェア	0272-21-2721	◎
8/5(日)	姫路	星電社姫路本店X68000フェア	0792-88-1717	◎
8/5(日)	高知	シャープ見体験フェアイン高知	0888-83-5522	※
8/11(土)・12(日)	宇都宮	計測技研新装開店フェア	0286-22-9811	◎
8/12(日)	伊勢崎	パソコンランド21伊勢崎店X68000フェア	0270-21-3121	◎
8/12(日)	東京	T-ZONE X68000フェア	03-257-2650	◎

◎印の会場で音遊サウンドライブを開催します。※印の会場には山下章氏来場。

シャープ株式会社

●お問い合わせは…シャープ(株)電子機器事業本部システム機器営業部
〒545大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)

'90 7月27日(金) 新規OPEN!!

CG画像制作センター 秋葉原サテライトオフィス

●新住所

〒101 東京都千代田区外神田6-3-8 外神田田島ビル3F

TEL 03-839-8481（但し、7月20日より通話可能）（JR秋葉原駅徒歩5分 地下鉄銀座線末広町駅徒歩2分）

——アンス・コンサルタンツ東京本部事務所（現高輪）は7月15日をもって上記へ移転します。——

●オープニング見学会

'90 7月27日(金) 11:00〜14:00 お取り引き先、マスコミ、他一般
15:00〜20:00 ユーザー様

主な業務案内／CG画像制作プロデュース・アプリケーション開発受託・サイクロンユーザー会ネットワークサポート・3次元CAD×CGシステム導入コンサルタント及び教育・アウトプットサービス等々



制作スタッフ募集!! CG画像制作センター

CGプロダクション（仮称：アトリエ68）
として、CG制作ユーザー会・関東支部を開設します。
ユーザーの方はどしどし制作スタッフ登録をして下さい。

※申し込み方法その他詳しくは福岡本社までお問い合わせ下さい

'90 オ2回サイクロンCG大会 9月24日に決定!!

全サイクロンシリーズユーザー対象〈98、68、TOWNS〉
静止画、アニメその他サイクロンを使用した作品なら何でもOK!!

- **作品受付期間**　8月10日〜9月8日（当日消印有効）
- **賞金・グランプリ**　20万円、その他賞金・賞品多数

※詳細は近日発表!!

サイクロンExpressα（エクスプレスアルファ）好評発売中!!

●サイクロンExpressα68 ……………… 98,000円
（SHARP X68000）

★CG大会には、αで応募しよう!!

お知らせ サイクロンテクニカルセミナー in 大阪
大阪シャープOAショールームにて開催中。
お申し込みはアンスまで。

★ステップ3　7月26日(木)
「ポリゴンを使用する」Z'S TRIPHONY DIGITAL CRAFTとのリンク
…………………………………………… 5,000円

★ステップ4　8月23日(木)
「絵を貼りつける」マッピングの使用法 ………… 5,000円



株式会社アンス・コンサルタンツ

九州本社／〒810 福岡市中央区平丘町68
phone.092-522-6347　FAX092-521-0400

X68000

The File Professor

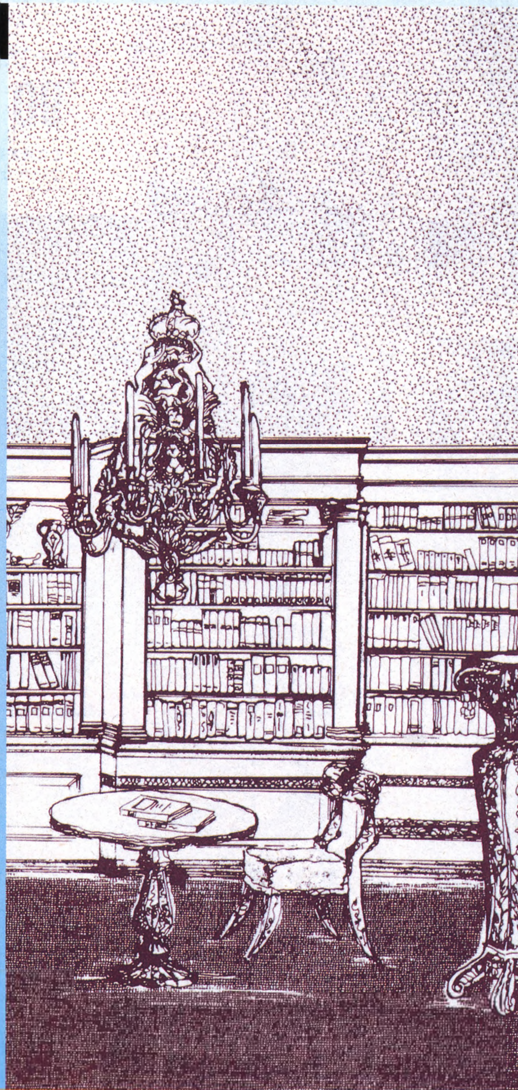
本格的ファイルマネージングソフトウェア

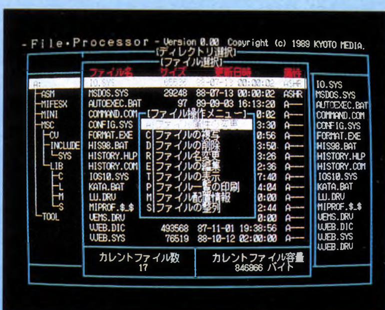
業界の新星、ロゴスシステムがユーザーの希望を1つの形にしました。これは必要だとか便利じゃない、快感だ！

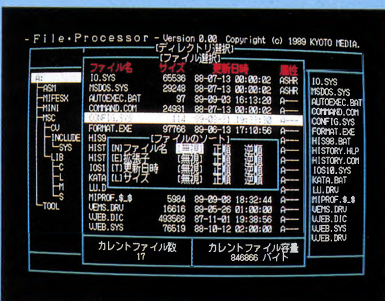
全国有名パソコンショップでお求め下さい。電話1本での通信販売も受付いたしております。

THE FILE PROFESSORの実力

ディスクのバックアップ、ディスクのエディット、ディスクの初期化、ディスクの比較、ディスクの検査、ディスクの情報、ＦＡＴのエディット、ファイルの検索、ディレクトリのコピー、ディレクトリの削除、ヴォリュームラベルの設定、ディレクトリの作成、ディレクトリ構造の再読み込み、ディレクトリ構造の印刷、ディレクトリ名の変更、ディレクトリ内容のソート、削除ファイルの復元、ファイル属性の変更、ファイルのコピー/移動、ファイルの削除、ファイルのエディット、ファイルの配置情報、ファイル一覧の印刷、ファイル名の変更、ファイルのソート、ファイル更新日時の変更、ファイルの表示、ファイルの奨行、カレンダー、ハードディスクの直撥エディット、システム情報の表示、コマンドシェル、現在時刻の変更。







メニュー選択方式を実現!!
初心者でも簡単に使える
（画面写真は、98用を開発中のものです。）

ロゴスシステム

このソフトはロゴスシステムのデビュー作です。でも、だからといってなめてもらっちゃぁ困ります。私達は、いろいろなソフトを作りました。そのどれもが他社から発売されていました。出来る事ならば自分達で発売したい！その願いがやっとかないました。

好評発売中！

定価28,000円

ロゴスシステム

〒615 京都市右京区西院上今田町17-1 L&Pビル4F
TEL（075）812-6383　　FAX（075）822-6915

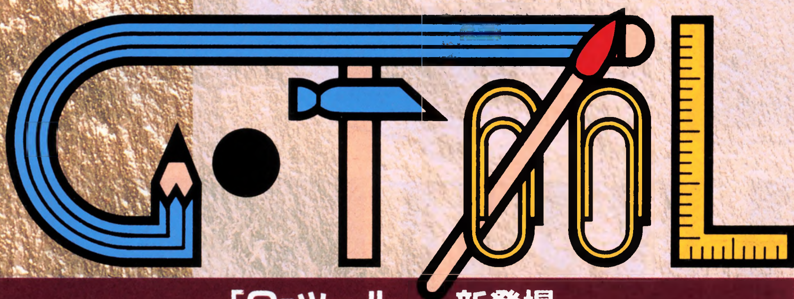
ザインの彩先端。

ニューコンセプトのアートキャンバス「G=ツール」、登場。

X68000ユーザーのクリエイティブマインドに火をつける新感覚のグラフィックツール。これまでのエディタ概念を払拭し、作品に挑むうえで必要不可欠なグラフィックキャラクタ・背景作成のすべてを備えたトータルツールです。ゲームデザインをはじめとしたオリジナルコンピュータアートが驚くほど自由に描けます。今回はグラフィックやスプライトのキャラクタの作成を目的とした「GR EDITモード」をご紹介します。

GR EDITモード





「G=ツール」 新登場。

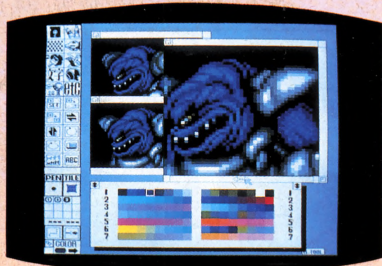
新発売
¥28,000

マルチウィンドウシステム：最大12枚まで描画ウィンドウが開ける優れたシステム環境を装備。複数のグラフィック・キャラクタが同時に作成できます。

ユーザーアイコンシステム：パレットやタイル、ペンなど、メインアイコン内の機能を使い勝手に合わせて、自分流のアイコン作成が可能。いちいちポップアップメニューを呼び出す必要もなくアートワークがはかどります。

マウス定義機能システム：マウスの左右クリックボタンに機能定義が可能。たとえば左利きの方もスムーズにオペレーティングできます。

高速メニューウィンドウ処理：メニューウィンドウの開閉は瞬時に。ユーザーアイコンシステムとの併用で、スピーディに仕事が進みます。



ZAINSOFT
株式会社ザイン・ソフト
〒676兵庫県高砂市米田町米田1162-1
TEL.(0794)31-7453

資料、オートデモ
請求券
Oh!X 8月号

KONAMI

世にも楽しいシューティングパズル

クォース™

QUARTH

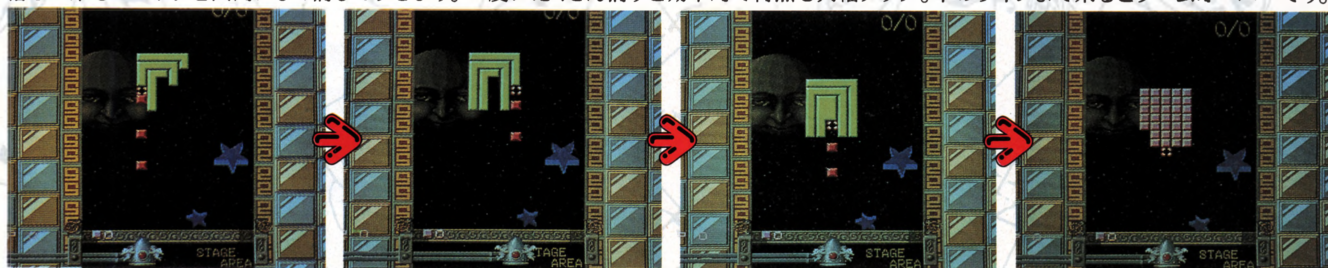
©KONAMI 1990

X68000版
好評発売中
6,800円(税別)

MSX2版 好評発売中 5,800円(税別)

PC-9801版 近日発売予定

落ちて来るブロックを四角にして消してゆきます。一度にたくさん消すと効率的で得点も大幅アップ。下のラインまで来るとゲームオーバーです。



コナミ株式会社

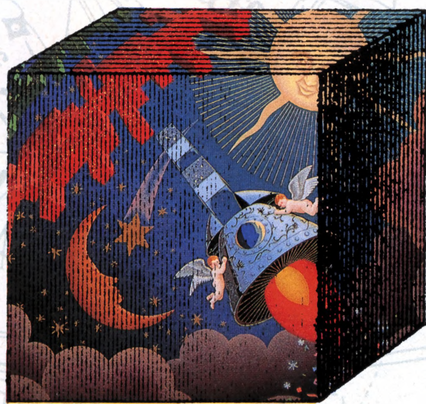
本社／〒101 東京都千代田区神田神保町3丁目25　大阪支店／〒561 大阪府豊中市庄内栄町4丁目23番18号　札幌営業所／〒060 札幌市中央区北1条西5丁目2番9号　福岡営業所／〒810 福岡市中央区天神2丁目8番30号



歴史に残る 前人未踏の 四角い宇宙

だれもが夢中になれるゲームを創りたい。

シンプルでいて奥が深い。だれでも気軽に遊べて、いつまでも飽きない。そんなピュアな、ほんとうの意味でのゲームがしたいと思うことがある。ゲームに対する熱い想いをもう一度じっくりと見つめて今、コナミが新たに発進する、楽園ゲームプロジェクト「クォース」。シューティングの楽しさと、パズルの思考性がマッチングした、すでにゲームセンターでは爆発人気の極楽行き超ソフトだ。ほら、もう引力がココロをズルズルと吸いこんでいる。君も、友も、父も、母も、老若男女を巻きこんで、楽園へ行こう。

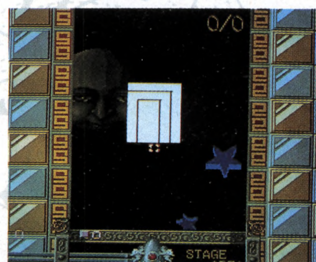




協力2Pで、仲直りもできる、ますます熱中の親切設計。



対戦2Pは、敵と相手の両方と戦う恐怖のケンカバトルだ。



アイテムブロックが出るとラッキー。

週刊テレホンサービス実施都市 【北海道地区】札幌011(241)4900 【東北地区】青森0177(22)5731 秋田0188(24)7000 【関東地区】東京03(262)9110 【北陸地区】新潟025(229)1141
【関西地区】大阪06(334)0399 【四国地区】松山0899(33)3399 【九州地区】福岡092(715)8200 大牟田0944(55)4444 志布志0994(72)0606 鹿屋0994(44)3977 以上、全11局開設しております

アーケード版
ジェミニウイング
待望の移植を実現!

ゲームセンターを賑わした
大人気シューティングゲーム
「ジェミニウイング」が、
キミのX68Kで今、蘇る!!



幾千の流星が降りそそいだ年、世界は蟲に覆われていた。人々は孤立し、街は滅び、植物に埋め尽くされた。蟲たちはさらに勢いを増し、残された僅かな地さえも蝕んでゆく。そして、ついに最高機密指令第307号、コード名ジェミニウイングは発動された……!

◆特徴◆

- ●二人同時プレイ可能
- ●MIDI対応(※)
  対応楽器 ローランド MT-32
  CM-32L CM-64
  (※)対応機種ごとに、それぞれ違ったBGMをお楽しみいただけます
  (※初期のMT-32では正常に演奏できません。)
- ●FM音源、ADPCM対応
- ●ジョイスティック対応
- ●5"2HD 2枚組

X68000 全シリーズ対応

標準価格 8,800円

Copyright c1987TECMO

闇の血族

THE PREDESTINED HOMICIDES #1

[redacted]

艶やかなファッション界を襲う奇怪な連続殺人事件。
南米の血に隠された秘密とは？
そして魅由を待ち受ける血族の宿命は？

NOV

上巻



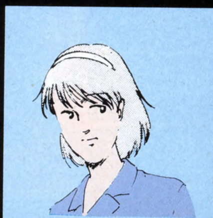


MIDI対応





[redacted]
（いざわ みゆ）
[redacted]



[redacted]
（ひめの りさ）
[redacted]



[redacted] リーン
[redacted] リーン）
[redacted]
体　重:53kg



東芝EMIより
「38万キロの虚空」CD
新発売　MT税込価格 2,250円
CD税込価格 2,530円

ノベルウェアシリーズ
「38万キロの虚空」
PC-9801 X68000 FM-TOWNS
各9,800円

好評発売中!!
メタルサイト
×68000　8,800円

※標準価格には消費税は含まれておりません。



株式会社 システム サコム
〒130 東京都墨田区両国4-38-16
両国桜井ビル4F
ハードウェア部 03(635)5145
ソフトウェア部 03(635)7609



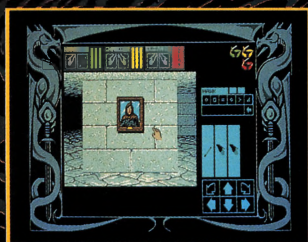
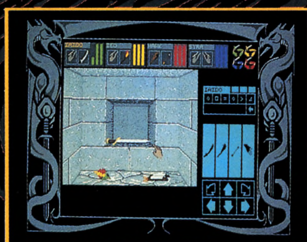
RPGの概念を一変させた傑作!

1988年発売と同時に世界中のゲーム・フリークを熱狂させた、あの「ダンジョン・マスター」が今、日本中を荒しまわる。
3Dグラフィックスによる複雑な迷路、数々の謎、パーティーを突然襲って来るモンスター。
そしてなによりもプレイヤーの考えること、見ること、手にすること、
すべてにリアルタイムで動いていく……本当の意味のリアルRPGだ。

なぜ、世界をそして日本をこれ程までに興奮させたのか!
その答えは君自身で出して欲しい。

もう逃げられない!

Dungeon ダンジョン・マスター Master



※画面写真はX-68000版





好評発売中	■X68000 マウス対応	■PC-9801VM21/11, VX, RX, RS, RA ■PC-98DO ■PC-9801UV21/11, UX, CV, EX, ES 要バス・マウス/アナログRGB対応	各¥9,800(税抜)

Produced by FTL Games © 1987, 1990 Software Heaven, Inc. © 1990 VICTOR MUSICAL INDUSTRIES, INC.

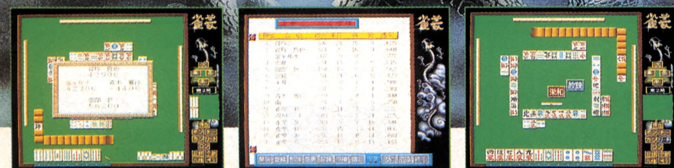
これが進化した麻雀ソフト、待望のX-68対応発売。

雀豪2 強知能版

麻雀ソフトの決定版登場！プレイすればするほど個性をもったプレイヤーに成長する自己成長型サンプリング機能と、より強化された推論型人工知能の搭載で限りなく実戦麻雀に近づいた。
リアルな４人囲みと見やすい麻雀牌、迫力ある効果音などの採用が麻雀ソフトの金字塔の地位を不動のものにする。

■8月上旬発売：X-68000 ■好評発売中：PC-9801シリーズ

各¥9,800(税抜き)



※画面写真はX-68版の開発画面です

■発売 ビクター音楽産業株式会社

通信販売 当社の商品をお近くのパソコンショップでお買い求めになれない場合、商品名、機種名、住所、氏名、電話番号を明記のうえ、下記住所まで定価プラス3%消費税分を現金書留にてお申し込み下さい。(送料無料) 〒151 東京都渋谷区千駄ヶ谷2-8-16 ビクター音楽産業㈱〈通信販売係〉

X68000

X68000の本質。「黒衣の貴公子」が今解き明かす。



©1990 T&E SOFT

黒衣の貴公子



neXt
RPG・ACT・SLG、最強のラインナップで
次世代体験……neXt!



●X68000 5"2HD 3枚組
- 全グラフィック書き起こし(高解像グラフィック 512×512ドット)
- ジョイスティック対応
- FM音源8音+ADPCM音源対応

●PC-9801VM、UVシリーズ PC-286、386シリーズ、NOTE対応
5"2HD/3.5"2HD 2枚組 ●サウンドボード対応 ●ジョイスティック対応
※VM、UVはRAM容量640Kバイト以上必要です。
※PC-9801/E/F/M/VF/U2/XAでは、ドライブ、RAM等の増設の如何にかかわらず、作動いたしません。
また1ドライブのみ搭載のPC-286L/LE/LFおよびPC-286NOTE Executiveでは、
ドライブを増設しても作動いたしません。
※高解像度(640×400ドット)カラーディスプレイをお使いください。液晶ディスプレイにも対応しています。

●PC-8801SRシリーズ・VA、98DO対応 5"2D 5枚組
- サウンドボードII完全対応、ADPCMをフルサポート ●ジョイスティック対応(98DOを除く)
- NEC純正128KRAMボード、I/Oデータ機器製RAMボードに対応したキャッシュドライバー搭載

●MSX2 MSX2+ (RAM64K以上、VRAM128K以上) 3.5"2DD 3枚組
- MSX-MUSIC(FM音源)対応 ●ジョイスティック対応

標準価格 各¥8,800
※表示価格に消費税は含みません

X68000版
7/13 FRI 新発売

RPG-neXt……ルーンワース 黒衣の貴公子
ACT-neXt……幻 獣 鬼
SLG-neXt……遥かなるオーガスタ



■通信販売ご希望の方は現金書留で料金と商品名・機種名と電話番号を明記の上、当社宛お送りください。(速達希望の方は300円プラス)
■カタログご希望の方は、送料として切手200円分を同封の上、カタログ請求券をお送りください。(葉書での請求はお断わりします)
●T&Eの最新情報がわかるテレフォンサービス 名古屋(052)776-8500

T&E SOFT
企画・開発・製造・販売
株式会社 ティーアンドイーソフト
〒465 名古屋市名東区豊が丘1810番地 PHONE:052-773-7770

カタログ請求券 Oh!x 8月号

超低金利クレジットOK

またまた秋葉原でおなじみの

注目!!

冬のボーナス一括払い
手数料(金利)無料
(平成2年12月末支払いをご利用下さい。)

7/15〜8/15

●お近くの方はお
●本体単品で特
●ビジネスソフト定

モデム(AIWA) 50台限定 (送料¥1,000)
PV-A24MNP5 (定価¥54,800)
●MNPクラス5
●2400bps
限定特価¥26,500 (送料・消費税込¥28,325)

CYBER STICK
●CZ-8NJ2
(定価¥23,800)
超特価!!
¥18,500 (送料・消費税込み¥19,570)



X68000シリーズ専用 特価¥16,480
MIDIインターフェースボード
SX-68M (サコム)
(純生コンパチ)定価¥19,800
送料・消費税込み!



X-1ターボZIII 特別ご提供品!! 台数限定
●CZ-888C＋CZ-860D＋M-2HD(10枚)
定価¥269,600▶特価¥164,800
(ボーナス併用も有りますTEL下さい)
・ジョイカード
・ゲーム3種
・パソコンラックⒶ3段
プレゼント中
送料消費税込み!!

12回	24回	36回	48回	60回
14,400	7,600	5,300	4,100	3,400

ジョイスティック 送料¥500
●X-1PRO
定価¥9,500▶特価¥7,800
●ASCII STICK
定価¥6,800▶特価¥5,500

P&A超低金利クレジットをご利用ください!!

NEW X68000EXPERT II/II-HD & PRO II/PRO II-HD & SUPER-HD (送料・消費税込)



EXPERT II

セットでお買い上げの方に、
●ディスケット10枚
●ゲーム3種
●ジョイカード2ケ
プレゼント中!!

EXPERT II

セット	構成	定価	特価	12回	24回	36回	48回	60回
Ⓐ	CZ-603C＋CZ-604D	¥432,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑ	CZ-603C＋CZ-605D	¥453,000	(価格はお電話下さい。)	30,200	15,900	11,000	8,500	7,100
Ⓒ	CZ-603C＋CZ-613D	¥473,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓓ	CZ-603C＋CU-21HD	¥486,000	(価格はお電話下さい。)	?	?	?	?	?

EXPERT II-HD

セットでお買い上げの方に、
●ディスケット10枚
●ゲーム3種
●ジョイカード2ケ
プレゼント中!!

EXPERT II-HD

セット	構成	定価	特価	12回	24回	36回	48回	60回
Ⓐ	CZ-613C＋CZ-604D	¥542,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑ	CZ-613C＋CZ-605D	¥563,000	(価格はお電話下さい。)	37,700	19,800	13,700	10,600	8,900
Ⓒ	CZ-613C＋CZ-613D	¥583,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓓ	CZ-613C＋CU-21HD	¥596,000	(価格はお電話下さい。)	?	?	?	?	?



PRO II

セットでお買い上げの方に、
●ディスケット10枚
●ゲーム3種
●ジョイカード2ケ
プレゼント中!!

PRO II

セット	構成	定価	特価	12回	24回	36回	48回	60回
Ⓐ	CZ-653C＋CZ-604D	¥379,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑ	CZ-653C＋CZ-605D	¥400,000	(価格はお電話下さい。)	26,800	14,100	9,700	7,600	6,300
Ⓒ	CZ-653C＋CZ-613D	¥420,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓓ	CZ-653C＋CU-21HD	¥433,000	(価格はお電話下さい。)	?	?	?	?	?

PRO II-HD

セットでお買い上げの方に、
●ディスケット10枚
●ゲーム3種
●ジョイカード2ケ
プレゼント中!!

PRO II-HD

セット	構成	定価	特価	12回	24回	36回	48回	60回
Ⓐ	CZ-663C＋CZ-604D	¥489,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑ	CZ-663C＋CZ-605D	¥510,000	(価格はお電話下さい。)	34,100	17,900	12,400	9,600	8,100
Ⓒ	CZ-663C＋CZ-613D	¥530,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓓ	CZ-663C＋CU-21HD	¥543,000	(価格はお電話下さい。)	?	?	?	?	?

■P&A恒例サマー大バーゲン祭
開催中!!
◎電話にて、ドンドンお問合せ下さい!!
クレジット表には、出せないほどの価格です。
メーカーさん、ご免なさい。
ユーザーの方には大歓迎されそうです。
今がチャンスです、ハイ。

SUPER-HD

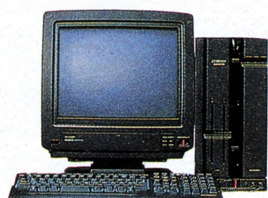
セットでお買い上げの方に、
●ディスケット10枚
●ゲーム3種
●ジョイカード2ケ
プレゼント中!!

SUPER-HD

セット	構成	定価	特価	12回	24回	36回	48回	60回
Ⓐ	CZ-623TN＋CZ-604D	¥592,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑ	CZ-623TN＋CZ-605D	¥613,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓒ	CZ-623TN＋CZ-613D	¥633,000	(価格はお電話下さい。)	42,700	22,500	15,500	12,100	10,100
Ⓓ	CZ-623TN＋CU-21HD	¥646,000	(価格はお電話下さい。)	?	?	?	?	?

X68000シリーズ 〜P&Aスペシャルセット＝限定誌上販売!!

台数限定 送料、消費税込み
セットでお買上げの方に、
●ディスケット10枚 ●ゲーム3種 ●ジョイカード2個 プレゼント中



機種	構成	定価	特価
EXPERT	CZ-602C＋CZ-612D	¥475,800	¥306,000
	CZ-602C＋CZ-604D	¥450,800	¥300,000
	CZ-602C＋CZ-605D	¥471,000	¥320,000
	CZ-602C＋CZ-613D	¥491,000	¥336,000
	CZ-602C＋CU-21HD	¥504,000	¥338,000
EXPERT-HD	CZ-612C＋CZ-612D	¥585,800	¥375,000
	CZ-612C＋CZ-604D	¥560,800	¥369,000
	CZ-612C＋CZ-605D	¥581,000	¥386,000
	CZ-612C＋CZ-613D	¥601,000	¥403,000
	CZ-612C＋CU-21HD	¥614,000	¥407,000
PRO-HD	CZ-662C＋CZ-612D	¥527,800	¥339,000
	CZ-662C＋CZ-604D	¥502,800	¥333,000
	CZ-662C＋CZ-605D	¥523,000	¥352,000
	CZ-662C＋CZ-613D	¥543,000	¥368,000
	CZ-662C＋CU-21HD	¥556,000	¥372,000



●本広告の掲載の商品の価格については、消費税は含まれておりません。 ●営業時間＝平日AM10:00〜PM7:00、日祭AM10:00〜PM6:00

1回～84回払いまでOK!!

★頭金なし! ★即日発送

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

超特価でクレジットが組める!!

X68000用ソフトコーナー(送料1ヶ～5ヶまで¥500)

Z's STAFF PRO68K Ver2.0(ツァイト)	定価¥ 58,000	特価¥ 39,700
Z's TRIPHONY デジタルクラフト(ツァイト)	定価¥ 39,800	特価¥ 29,300
テラッツォ(ハミングバード)	定価¥ 19,800	特価¥ 15,800
KAMIKAZE(サムシング・グッド)	定価¥ 68,800	特価¥ 46,000
EW&EI(イースト)	定価¥ 38,800	特価¥ 28,800
C&Professional Pack(マイクロウェアジャパン)	定価¥ 58,800	特価¥ 43,000
Final Ver3.2(エーエスピー)	定価¥ 38,000	特価¥ 30,000
DATA PRO68K CZ220BS	定価¥ 58,000	P&A特価
CARD PRO68K CZ226BS	定価¥ 29,800	TEL下さい!
C compiler PRO68K CZ211LS	定価¥ 39,800	特価¥ 32,000
OS-9/X68000 CZ219SS	定価¥ 29,800	P&A特価 TEL下さい。
AI-68K CZ234LS	定価¥ 188,000	特価¥ 143,000
THE 福袋 V2.0 CZ224LS	定価¥ 9,900	特価¥ 7,700
SOUND PRO68K	定価¥ 15,800	特価¥ 12,500
MUSIC PRO68K CZ213MS	定価¥ 15,800	P&A特価 TEL下さい。
Sampling PRO68K CZ215MS	定価¥ 17,800	特価¥ 14,000
MUSIC-studio PRO68K 237MS	定価¥ 15,800	P&A特価 TEL下さい。
MUSIC-PRO68K〔MIDI〕247MS	定価¥ 28,800	特価¥ 22,000
New-print Shop 221HS	定価¥ 19,800	P&A特価
Communication 223CS	定価¥ 19,800	TEL下さい。
C-TRACE68 Ver.3.0(キャスト)	定価¥ 98,000	特価¥ 77,000
サイクロンEXPRESS α68	定価¥ 98,000	特価¥ 72,000
G68K Ver2 PRO	定価¥ 22,000	特価¥ 16,300
THE FILE PROFESSOR(ロゴシステム)	定価¥ 28,000	特価¥ 20,500
Gツール(ザインソフト)	定価¥ 28,000	特価¥ 20,500
たーみのる2(SPS)	定価¥ 17,800	特価¥ 13,500
マジックパレット(ミュージカルプラン)	定価¥ 19,800	特価¥ 14,900
Hyper word CZ-251BS	定価¥ 39,800	特価¥ 30,900

●ゲームソフト20%OFF OK!! (一部ソフト除く)

周辺機器コーナー(送料¥1,000)

Ⓐ CZ-8NSI	定価¥188,000	特価¥145,000
Ⓑ CZ-6VTI	定価¥ 69,800	特価¥ 54,000
Ⓒ CZ-6TU	定価¥ 33,100	特価¥ 25,000
Ⓓ BF-68PRO	定価¥ 19,800	特価¥ 15,500
Ⓔ CZ-6BEI	定価¥ 35,000	特価¥ 26,500
Ⓕ CZ-6BEIA	定価¥ 38,000	特価¥ 28,600
Ⓖ CZ-6BE2	定価¥ 79,800	特価¥ 60,000
Ⓗ CZ-6BE4	定価¥138,000	特価¥107,000
Ⓘ CZ-6BFI	定価¥ 49,800	特価¥ 38,200
Ⓙ CZ-6BPI	定価¥ 79,800	特価¥ 61,000
Ⓚ CZ-6BMI	定価¥ 26,800	特価¥ 20,300
Ⓛ CZ-6EBI	定価¥ 88,000	特価¥ 67,500
Ⓜ AN-S100	定価¥ 36,600	特価¥ 28,500
Ⓝ CZ-6SDI	定価¥ 44,800	特価¥ 35,000
Ⓞ CZ-8PC3	定価¥ 65,800	P&A超特価 TEL下さい。
Ⓟ CZ-8PC4	定価¥ 99,800	P&A超特価 TEL下さい。
Ⓠ CZ-8PG1	定価¥130,000	P&A超特価 TEL下さい。
Ⓡ CZ-8PG2	定価¥160,000	P&A超特価 TEL下さい。
Ⓢ CZ-8PK10	定価¥ 97,800	P&A超特価 TEL下さい。
Ⓣ CZ-6PVI	定価¥198,000	特価¥153,000
Ⓤ IO-735X	定価¥248,000	特価¥190,000
Ⓥ CZ-8BSI	定価¥ 23,800	特価¥ 19,000
Ⓦ PIO-6BE1-A(I/O DATA)	定価¥25,000	特価¥18,200
Ⓧ PIO-6BE2-2M(I/O DATA)	定価¥50,000	特価¥36,800
Ⓨ PIO-6BE4-4M(I/O DATA)	定価¥88,000	特価¥64,800

X68000用ハードディスク (送料¥1,000)

アイテム
- ●HXD-040(40MB/23ms)……定価¥118,000▶特価¥ 88,000
- ●HXD-042(増設用)……定価¥128,000▶特価¥ 95,000

アイテック
- ●ITX-640(40MB/28ms)……定価¥158,000▶特価¥101,000
- ●ITX-680(80MB/20ms)……定価¥198,000▶特価¥131,000

プリンター(ケーブル・用紙付)限定5台 新品(送料¥1,000)

- ●CZ-8PC3(カラー漢字24ドット熱転写プリンター)
  定価¥65,800……特価¥39,800
- ●CZ-8PK8(24ピン漢字プリンター136桁)
  定価¥152,000……特価¥69,000
- ●CZ-8PC4 P&A特選!!(カラー漢字48ドット熱転写プリンター)
  定価¥99,800……特価¥56,000

モデムコーナー (送料¥1,000)

Ⓐ MD-24FS5(オムロン)	定価¥ 49,800	特価¥ 34,800
Ⓑ MD-24FS7(オムロン)	定価¥ 64,800	特価¥ 45,000
Ⓒ コムスター2424/4(NEC)	定価¥ 38,800	特価¥ 28,000
Ⓓ コムスター2424/5(NEC)	定価¥ 44,800	特価¥ 32,000

P & A 特選パソコンラック (送料無料)移動自由(キャスター付)

- Ⓐ 3段 875(H)×580(D)×610(W) ¥9,000
- Ⓑ 4段 1320(H)×600(D)×630(W) ¥12,000
- Ⓒ 5段 1280(H)×600(D)×620(W) ¥15,000

中古パソコン 送料¥2,000

●X-68000セット	¥210,000	●CZ-856C	¥45,000	●CU-14AG2	¥30,000
●X-68000ACEセット	¥240,000	●CZ-870C	¥55,000	●CU-14H2	¥30,000
●X1ターボZセット	¥100,000	●CZ-881C	¥65,000	●CZ-8PC2	¥25,000
●X-1G/30セット	¥ 39,000	●CZ-820D	¥10,000	●CZ-8PK6	¥32,000
●CZ-822C	¥ 15,000	●CU-14GB	¥ 5,000		
●CZ-830C	¥ 25,000	●CU-14BD	¥25,000		

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK!!

■まずはお電話下さい。
03-651-1884 FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

●月々¥1,000円からOK!! ●ボーナス払いOK(夏冬10回までOK)
●支払い回数 1回～84回 ●お支払いは、8ヶ月先からでもOK!!

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブル etc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

●定休日/毎週水曜日＝第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～84回払いまで出来ます。但し、1回のお支払い額は¥1000円以上。

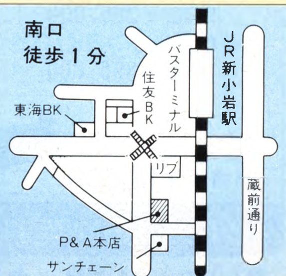
超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	2.5	3.5	5.0	5.0	9.0	10.5	14.5	19.0	24.5	32.0	38.5

●マイコン
●ビデオ
●ビデオテープ

P&A 株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-651-0148(代) FAX 03-651-0141

営業時間
平日:AM10:00～PM7:00
日祭:AM10:00～PM6:00

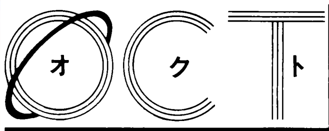


●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

■店頭にて、ゲームソフト25%OFF!!（税別）、超低金利オクトハッピークレジットをご利用下さい!!

■平成2年夏のボーナス一括払い（8月末）OK!!手数料ナシ!!おトクです。ぜひ!!超低金利クレジットをご利用下さい。

パソコンプラザ





'90 オクトで始まるパソコンワールド

☎03-730-6271

●営業時間 AM 11:00〜9:00/日曜・祭日PM7:00　電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

全国通販

●定休日毎週火曜日 祭日の場合翌日になります。

オクト ラクラククレジット

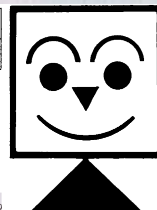
1回	2%	3回	2.5%	6回	3.5%	10回	5%	12回	5%	15回	7.5%
18回	9%	20回	10%	24回	11%	30回	14.5%	36回	15.5%	48回	20%

OCT-1 システム インフォメーション

- ▶全商品保証付（メーカー保証）
- ▶超低金利ハッピークレジット（1回〜60回）頭金ナシOK！
- ▶ボーナス一括払いOK！ボーナス2回払いOK!!
- ▶配達日の指定OK！（万全なサポート体制）
- ▶商品の組合せ自由！ オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト セレクテッドシステム

広告掲載商品以外の製品も取扱っております。





蒲田

●平成2年、8月末一括払い（手数料ナシ）OK!!
OKだよ〜ん。超低金利 ハッピークレジットですゾ
EXPERT II・PRO II 新発売記念セール開催中!!

OPEN

★下記セットでお買い上げの方にはプレゼント!! ●①MD-2HD 10枚②ジョイカード 2個（連射式）③シリコンキーボードカバー

お好みのセットをお選び下さい。

送料無料

●SX-WINDOW搭載。
●40Mバイトハードディスク搭載



EXPERT II・EXPERT II-HD

●CZ-603C-BK/GY 定価¥338,000
●CZ-613C-BK/GY 定価¥448,000

現金特価!! お電話下さい。

推選

●SX-WINDOW搭載。
●拡張I/Oポート4スロット装備



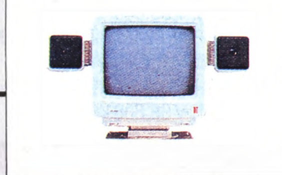
PRO II・PRO II-HD

●CZ-653C-BK/GY 定価¥285,000
●CZ-663C-BK/GY 定価¥395,000



CZ-8NJ2
●インテリジェントコントローラ
定価¥23,800
超特価¥18,800

15型カラーディスプレイTV



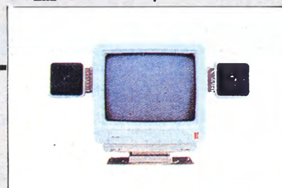
CZ-605D-GY/BK
定価¥115,000

15型カラーディスプレイTV



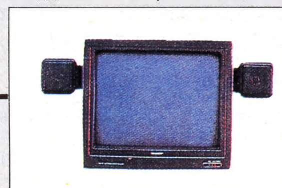
CZ-613D-GY/BK
定価¥135,000

14型カラーディスプレー



CZ-604D-GY/BK
定価¥94,8000

21型カラーディスプレイ



CU-21HD
定価¥148,000

Ⓐ CZ-603C＋CZ-605D……定価合計¥453,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓑ CZ-613C＋CZ-605D……定価合計¥563,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓒ CZ-653C＋CZ-605D……定価合計¥400,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓓ CZ-663C＋CZ-605D……定価合計¥510,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓔ CZ-603C＋CZ-613D……定価合計¥473,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓕ CZ-613C＋CZ-613D……定価合計¥583,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓖ CZ-653C＋CZ-613D……定価合計¥420,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓗ CZ-663C＋CZ-613D……定価合計¥530,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓘ CZ-603C＋CZ-604D……定価合計¥429,800▶オクト大特価

12回	¥28,000	24回	¥14,800	36回	¥10,200	48回	¥8,000

Ⓙ CZ-613C＋CZ-604D……定価合計¥542,000▶オクト大特価

12回	¥36,000	24回	¥19,000	36回	¥13,100	48回	¥10,200

Ⓚ CZ-653C＋CZ-604D……定価合計¥379,800▶オクト大特価

12回	¥25,400	24回	¥13,400	36回	¥9,300	48回	¥7,200

Ⓛ CZ-663C＋CZ-604D……定価合計¥489,800▶オクト大特価

12回	¥32,200	24回	¥17,000	36回	¥11,800	48回	¥9,200

Ⓜ CZ-603C＋CU-21HD……定価合計¥486,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓝ CZ-613C＋CU-21HD……定価合計¥596,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓞ CZ-653C＋CU-21HD……定価合計¥433,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓟ CZ-663C＋CU-21HD……定価合計¥543,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

♡どんどんTELしょう。安くなるかもヨ!!

♡クレジット価格は、消費税込みですヨ。ご利用下さい!!

※クレジットの回数は1回〜60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット：送料無料　●店頭デモ実施中…専門の係員が詳細にアドバイス致します。ぜひご来店下さい。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■店頭にて、ゲームソフト25%OFF!!(税別)、超低金利 ハッピークレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

■平成2年夏のボーナス一括払いOK!!(8月末)手数料ナシ!!

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の製品も取扱っております。

チャンス! X68000・SUPER-HD(チタン)=好評・発売中 どんどんTEL下さいネ。 送料¥2,000

SX-WINDOW搭載。



●ザ・ワークステーションと呼ぶにふさわしいスーパーな68000.!! 新登場.!! SUPER-HD。

※プレゼント! ①MD-2HD10枚 ②アフターバーナー(¥9,200) ③ジョイカード(連射式) ④シリコンキーボード(¥2,800)

X68000 SUPER-HD

●CZ-623C-TN+CZ-613D-TN
定価合計¥633,000…大特価.!! TEL下さい。

※マウス・トラックボール付.!! ディスプレイにはスピーカ2個、チルト台付.!!

他のディスプレイ① CZ-602D、② 612D、③ CZ-603D、④ CU-21HDの組合せもございますのでお問い合せ下さい。

12回	?	24回	?	36回	?	48回	?

♡安くてゴメンなさい。今だけヨ.!!

※超低金利クレジットご利用下さい。1回～60回払い、頭金ナシ! ボーナス1回払い、ボーナス2回払いOK!

X68000 EXPERT-HD

オクト限定スペシャルセット

ラストチャンス.!! 早い者勝ち.!!



- CZ-612C(BK) (¥466,000)
- CZ-602D(BK) (¥99,800)
- MD-2HD 10枚
- ジョイカード(連射式×2個)
- ゲーム

オクト超特価 ¥364,000(送料・消費税込み.!!)

※ディスプレイ=①CZ-604D ②CZ-605D ③CZ-613D ④CU-21HD との組合せもございます。TEL下さい。

オクト特選 シャープ周辺機器 (送料¥1,000)

- CZ-6BE1 IBM増設RAMボード……(¥35,000)▶特価¥26,500
- CZ-6BE1B 1MB増設RAMボード………(¥28,000)▶特価¥21,000
- CZ-6BE2 2MB増設RAMボード……(¥79,800)▶特価¥60,500
- CZ-6BE4 4MB増設RAMボード……(¥138,000)▶特価¥104,800
- CZ-6BF1 増設用RS-232Cボード……(¥49,800)▶特価¥38,500
- CZ-6BG1 GP-IBボード……………(¥59,800)▶特価¥45,000
- CZ-6BM1 MIDIボード………………(¥26,800)▶特価¥20,500
- CZ-6BN1 スキャナ用ハラレルボード‥(¥29,800)▶特価¥22,800
- CZ-6BP1 数値演算フロセッサボード(¥79,800)▶特価¥60,500
- CZ-6BO1 ユニバーサルI/Oボード…(¥39,800)▶特価¥30,500
- CZ-6EB1/BK 拡張I/Oボックス…………(¥88,000)▶特価¥66,800
- CZ-6VT1/BK カラーイメージ・ユニット…(¥69,800)▶特価¥53,000
- CZ-6BL1 LANボード………………(¥268,000)▶大特価
- CZ-8NM2A マウス………………(¥68,800)▶特価¥5,300
- CZ-8NT1 マウストラックボール‥(¥98,800)▶特価¥7,500
- CZ-8NS1 カラーイメージスキャナ……(¥188,000)▶大特価
- CZ-6BC1 FAXボード…………(¥79,800)▶特価¥60,500
- CZ-8TM2 モデムユニット………(¥49,800)▶特価¥38,000
- CZ-64H 増設ハードディスク…(¥120,000)▶大特価
- CZ-6TU GY/BK RGBシステムチューナー…(¥33,100)▶特価¥25,000
- BF-68PRO 高性能CRTフィルター……(¥19,800)▶特価¥15,500
- SX-68M(システムサコム) MIDIボード…………(¥19,800)▶特価¥15,000
- PIO-68BE1-A(I/O DATA) 1MB増設RAMボード……(¥25,000)▶特価¥18,500
- PIO-6BE2-2M(I/O DATA) 2MB増設RAMボード……(¥50,000)▶特価¥37,000
- PIO-6BE4-4M(I/O DATA) 3MB増設RAMボード……(¥88,000)▶特価¥65,000

オクト面白グッズ

アイテック(送料¥1,000)
- IT-X640(¥158,000)…………特価¥103,000
- IT-X680(¥198,000)…………特価¥134,000

モデムコーナー(送料¥1,000)

- MD-1200AIII…………特価¥14,800
- MD-24FS4…………特価¥31,500
- MD-24FS5…………特価¥34,800
- MD-24FP4…………特価¥27,900
- MD-12FS…………特価¥15,000

熱転写カラー漢字プリンター (ケーブル用紙付) 送料¥1,000

CZ-8PC4 ¥99,800 限定



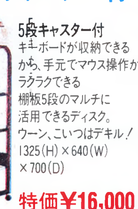
- 48ドット サーマルヘッド
- B5～B4まで
- ハガキ可能
- カラー対応

オクト大特価¥55,800

① CZ-8PC3(24ドット熱転写カラー漢字プリンター) 定価¥65,800……………特価¥45,000
② CZ-8PK9(24ピン漢字プリンター80桁) 定価¥89,800………大特価.!! TEL下さい。
③ CZ-8PK10(24ピン漢字プリンター136桁) 定価¥97,800………大特価.!! TEL下さい。
④ CZ-8PG1(24ピンカラー漢字プリンター80桁) 定価¥130,000………大特価.!! TEL下さい。
⑤ CZ-8PG2(24ピンカラー漢字プリンター136桁) 定価¥160,000………大特価.!! TEL下さい。
⑥ IO-735×(カラーイメージジェット) 定価¥248,000………大特価.!! TEL下さい。

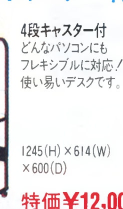
パソコンラック 推奨 送料無料

①五段キャスター付

5段キャスター付
キーボードが収納できるから、手元でマウス操作がラクラクできる
棚板5段のマルチに活用できるディスク。
ウーン、こいつはデキル!
1325(H)×640(W)×700(D)
特価¥16,000

②四段キャスター付

4段キャスター付
どんなパソコンにもフレキシブルに対応!
使い易いデスクです。
1245(H)×614(W)×600(D)
特価¥12,000

③三段キャスター付




3段キャスター付
場所を選ばない簡易で便利なディスクです。
限定
1175(H)×640(W)×600(D)
特価¥8,800

X68000ソフト大セール実施中※ゲームソフトオール25%off

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0 (シャフト)定価¥58,000 オクト特価¥40,000
〈データーベース〉●KAMIKAZE (サムシンググッド)定価¥68,000 オクト特価¥46,000
〈グラフィック〉●C-TRACE68 (キャスト)定価¥68,000 オクト特価¥51,000
〈C言語〉●C & Professional Pack (マイクロウェアジャパン)定価¥58,000 オクト特価¥44,000
〈グラフィック〉●サイクロン エキスプレス 定価¥78,000 オクト特価¥58,000
〈グラフィック〉●デジタルクラフト 定価¥39,800 オクト特価¥28,000
〈ワープロ〉●ハイパーワード 定価¥39,800 CZ-251BS オクト特価¥29,800

型名	商品	定価	特価
CZ-211LS	Ccompiler PRO-68K	¥39,800	¥28,800
CZ-212BS	BUSINESS PRO-68K	¥68,000	¥48,000
CZ-213MS	MUSIC PRO68K	¥18,800	¥13,500
CZ-214MS	SOUND PRO-68K	¥15,800	¥11,500
CZ-215MS	Sampling PRO-68K	¥17,800	¥12,800
CZ-219SS	OS-9/X68000	¥29,800	¥21,000
CZ-220BS	DATA PRO-68K	¥58,000	¥41,000
CZ-221HS	New Print Shop PRO-68K	¥19,800	¥14,300
CZ-223CS	Communication PRO-68K	¥19,800	¥14,300
CZ-224LS	THE 福袋 V2.0	¥9,900	¥7,500
CZ-226BS	CARD PRO-68K	¥29,800	¥21,300
CZ-241BS	システム手帳リフィル集	¥9,800	¥7,500
CZ-242BS	活用フォーム集	¥9,800	¥7,500
CZ-244SS	Homan 68K Ver.2.0	¥9,800	¥7,500
CZ-247MS	MUSIC PRO-68K(MIDI)	¥28,800	¥20,800
CZ-240BS	Stationery PRO-68K	¥14,800	¥11,500
CZ-243BS	CYBER NOTE PRO-68K	¥19,800	¥15,200
EW		¥38,000	¥29,800
G-68K		¥14,800	¥11,400
E-68		¥19,800	¥15,300

★オクト今月だけの新品限定販売(各1台限) (送料¥1,000)

- CZ-822C(BK)定価¥ ? 大特価¥18,800
- CZ-888C(BK)定価¥168,000 大特価¥69,800
- CZ-601C(BK)定価¥319,800 大特価¥174,000
- CZ-611C(BK)定価¥399,800 大特価¥218,000
- CZ-652C(BK)定価¥298,000 大特価¥188,000
- CZ-662C(BK)定価¥408,000 大特価¥248,000
- CZ-601D(BK)定価¥119,800 大特価¥68,000
- CZ-601D(GY)定価¥119,800 大特価¥68,000
- CZ-612D(GY)定価¥119,800 大特価¥74,000

店頭ゲームソフトオール25%off! ビジネスソフト25%より特価中

●尚、送料として1ケ¥500、2ケ¥700、3ケ以上で¥1,000となります。(税別)

超低金利クレジットをご利用下さい。

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

1回	2%	3回	2.5%	6回	3.5%	10回	5%
12回	5%	15回	7.5%	18回	9%	20回	10%
24回	11%	30回	14.5%	36回	15.5%	48回	20%

振込先
富士銀行 久ヶ原支店 ㊌No.1824
三菱銀行 蒲田支店 ㊌No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※連休のお知らせ=7/31(水)、8/1(水)は連休です。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

掲載商品2万円以上
送料無料!!

夏ツクモ! ザバーゲン! 決算セールは7/31(火)迄です。

冬のボーナス一括払受付中!くわしくはお問い合せ下さい。

ツクモ決算!展示棚ズレ品

SHARP PA-6500	SHARP PA-7000	SHARP CZ-8PC3	SHARP CZ-8PK7	SHARP CZ-8PK8
定価¥17,800	定価¥19,800	定価¥65,800	定価¥122,000	定価¥152,000
限定3台	限定9台	24ドット熱転写カラー漢字プリンター	24ピン、80桁 限定5台	24ピン、136桁 限定5台
55% OFF	51% OFF	60% OFF	51%OFF	45%OFF
決算特価¥9,800	決算特価¥9,800	決算特価¥39,800	決算特価¥59,800	決算特価¥83,800

ツクモ通販受注センターフリーダイヤル
0120(377)999

商品のお問い合せは各店又は通販部☎03(251)9911へ

NEW

PROII CZ653C 定価¥285,000 / CZ663C 定価¥395,000
●次世代のインテリジェンス、SX-WINDOW搭載 ●知的ニュースタンダードフォルム ●BIOSの改良によりハイスピード処理を実現 ●2Mバイトの大容量メモリを標準装備 ●拡張I/Oポート4スロット標準装備

EXPERTII CZ603C 定価¥338,000 / CZ613C 定価¥448,000
●次世代のインテリジェンス、SX-WINDOW搭載 ●象徴のフォルム、マンハッタンシェイプ ●BIOSの改良によりハイスピード処理を実現 ●3Mバイトの大容量メモリを標準装備

SUPER HD CZ623C 定価¥498,000
●次世代のインテリジェンス、SX-WINDOW搭載 ●「チタン」カラーのクォリティブラック ●80MBハードディスク搭載 ●世界標準SCSIインターフェース標準装備 ●BIOSの改良によりハイスピード処理を実現 ●3Mバイトの大容量メモリを標準装備

LET'S MUSIC♪♬

Aセット		Bセット	
CM-32L	¥69,000	CM-64	¥129,000
SX-68M	¥19,800	SX-68M	¥19,800
Musicstudio Mu-1	¥19,800	Musicstudio Mu-1	¥19,800
合計定価	¥108,600	合計定価	¥168,600
ツクモ特価	¥91,800	ツクモ特価	¥144,000
(消費税別途¥2,754)		(消費税別途¥4,320)	
クレジット例(税込)月々¥5,830×18回払		クレジット例(税込)月々¥7,107×24回払	

★Musicstudio PRO-68K V1.1又は、Music PRO68K(MIDI)のソフトの場合には¥8,000プラスになります。

X68000用ハードディスク

シャープ
光磁気ディスクユニット
CZ-6MO1 予約受付中!
SCSIボード
CZ-6BS1 予約受付中!

アイテック
IT X640 定価¥158,000 特価¥89,800
IT X680 定価¥198,000 特価¥118,000

Software tools

GRAPHIC TOOLS
マジックパレット	特価¥16,830
Z's STAFF PRO-68K	特価¥49,300
サイクロンExpressα68	特価¥83,300
デジタルクラフト	特価¥33,800

電子手帳ソフト
CYBERNOTE PRO-68K	特価¥16,830
Stationery PRO-68K	特価¥12,580
※通信ケーブル CE-300L	特価¥2,520

電子手帳 & ポケコン

PA-8600	特価¥24,800
PA-7500	特価¥17,800
PC-E500	特価¥24,800

情報ツール



All in Note

●「Business Mate」標準装備
●20MバイトHD搭載
●フリートップサイズ
●小さいボディに高性能

周辺機器
3.5インチフロッピーディスクドライブ
UE-1F04 定価¥49,800
一体型外部バッテリ
UE-1X07 定価¥26,000

表計算ソフト
Microsoft EXCEL Ver.2.1 定価¥98,000

ワープロソフト
一太郎AX 定価¥68,000
書院AX(UE-6Z10) 定価¥49,800

AX286N-H2
定価¥398,000

★発売記念特別価格にて提供中!! 詳しくはお電話で!

通信モデム & ソフト

アイワ
PV-A24MNP5 ツクモ特価¥29,800 (消費税別途¥894)
た〜みのる2 ツクモ特価¥15,000 (消費税別途¥450)

X68000用メモリーボード

一流メーカー
1MB増設用RAMボード ツクモ特価¥19,800
(ACE&PROシリーズ内蔵用1MB)
2MB増設用RAMボード 定価¥50,000 ツクモ特価¥42,500
4MB増設用RAMボード 定価¥88,000 ツクモ特価¥74,500
※2MBと4MBは全てシリーズ対応拡張スロット用

ツクモグローバルカード

18才以上なら学生でもOK!

国内・外で活躍!
使って便利、持ってて安心!ツクモグローバルカードはジャックス・VISA、セントラル・マスターとの提携カードです。ツクモ各店でのお買物がらくらくできるうえに、国内はもとより海外でのショッピングもOK!しかも18才以上なら学生でもOK!

お申し込みは(03)251-9898又は各店で

秋葉原各店

(地図: 至御茶の水、昌平橋通り、5号店、7号店、万世橋、シンクタ電機本店、ニューセンター店、中央通り、秋葉原駅、至上野、至神田、JR山手・京浜東北線、至浅草橋)

㊟AM10:15〜PM7:00 ㊡毎週木曜日と8/15

★表示価格には消費税は含まれておりません。

ツクモは「スーパーX PRO SHOP」です。

PRO STAFF ツクモ

九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号

7号店 荒井
N C店 福地

ツクモ7号店 ☎03-253-4199(担当/荒井)

便利で安心な通信販売
通信販売部☎03-251-9911

■ニューセンター店 ☎03-251-0987(担当/福地)
■ツクモ5号店 ☎03-251-0531(担当/川名)
■名古屋1号店 ☎052-263-1655(担当/吉高)
■名古屋2号店 ☎052-251-3399(担当/横山)
■ツクモ札幌 ☎011-241-2299(担当/村井)

★商品のご注文は在庫確認の上お願いします。

カード払い	全国代金引き換え配達	クレジット払い	現金書留払い	銀行振込払い	各種リース払い
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通信販売部へお申し込み下さい。	お申し込みは☎03-251-9911へお電話1本!配達日の指定もできます。	月々¥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中!	〒101-91 東京都千代田区神田郵便局私書箱135号 九十九電機㈱通信販売部 oh!X係	事前に☎でお届け先をご連絡下さい。富士銀行 神田支店(普)No.894047 九十九電機㈱	くわしくは各店にお問い合せ下さい。ケースに合わせてご相談にのらせて頂きます。

表紙ぎゃらりぃ

1982年5月18日の創刊以来，本誌は誌名を変えても変わらぬ心で誌面作りを続けてきました。応援してくださった読者の皆さん本当にありがとうございます。おかげさまでOh! Xは通巻100号を数えることになりました。ここにその表紙のすべてをご紹介しましょう。これからも本誌をよろしくお願いいたします。

MZ専門誌としてデビューしたOh!MZ。創刊号は104ページで620円。あまりに高いとの声に次号から480円に値下げしたが……。ちなみに表紙はマジックバス，オークスターなるヒロインが活躍した。まだX1が誕生する前の時代である。

❶創刊号



❷7月号



❸8月号



❹9月号



❺10月号



❻11月号



❼12月号



パソコンテレビX1の登場で誌面に緊張感が。だが，誌名までが変わってしまう事態を予想した人はどれだけいたであろうか。時はMZ-700の全盛期。一時は読者の4割を超えることもあり，本誌は飛躍的な部数アップを記録した。

❽1月号



❾2月号



❿3月号



⓫4月号



⓬5月号



⓭6月号



⓮7月号



⓯8月号



⓰9月号



⓱10月号



⓲11月号



Oh!CZが大反響

⑲12月号



4月号からあのシド・ミードが表紙を飾る。増ページと共に内容も充実し，ほぼ現在のスタイルを確立。そして11月号には新製品X1turboの歴史に残る大特集が。MZユーザーの目がこれ以来反感から羨望へと変化したという。

⑳1月号



㉑2月号



㉒3月号



㉓4月号



㉔5月号



㉕6月号



㉖7月号



㉗8月号



㉘9月号



㉙10月号



㉚11月号



㉛12月号



感動のX1turbo特集

全機種共通システムS-OSがスタート。また，満開一号を発表(?)した祝一平氏が「試験に出るX1」を連載。時代はその筋へと流れていく。Oh!MZがユーザーと共にあるべきパーソナルコンピューティングを追求しだしたのはこのころだ。

㉜1月号



㉝2月号



㉞3月号



㊱4月号



㊲5月号



㊳6月号



パソコン史に残るS-OS発表

㊴7月号



㊵8月号



㊶9月号



42 10月号



43 11月号



44 12月号



35 ADVANCED MZ-700





本誌唯一の別冊。発売が遅れてMZ-700のユーザーをやきもきさせた。

読者参加を強く呼び掛ける特別企画「GAME OF THE YEAR」「言わせてくれなくちゃだワ」を開催。このままでは世のパソコンがすべて実務マシン一色になるという不安のなか，ついに夢のマシンX68000が衝撃のデビューを遂げたのだった。

45 1月号



46 2月号



47 3月号



48 4月号



49 5月号



50 6月号



51 7月号



52 8月号



53 9月号



54 10月号



55 11月号



X68000が初登場！

56 12月号



MZ-2861を機にMZグループがビジネスコンピュータへの路線転換，パーソナルユースはXfamilyに絞られる。そのため本誌は12月号でOh!MZ→Oh!Xと改題した。なお，1月号から翌年3月号までの表紙イラストは永沢しげる氏が担当。

57 1月号



58 2月号



59 3月号



60 4月号



61 5月号



62 6月号



63 7月号



64 8月号



65 9月号



66 10月号



67 11月号



68 12月号



これが初のOh!X

X68000ユーザーが増えるなか，本誌では創刊6周年企画として8TRON計画を発表，昭和70年代を目指した究極の8ビットパソコンの姿を考えた。結局70年は来なかったが……。また，4月号からは画家の松葉口忠夫氏に表紙絵を依頼。

69 1月号



70 2月号



71 3月号



72 4月号



73 5月号



74 6月号



きわどい内容が満載

75 7月号



76 8月号



77 9月号



78 10月号



79 11月号



80 12月号



名実共にパーソナルマシンの一大勢力に成長したX68000。読者の割合も半数に達し，誌面もX68000を中心にゲーム，グラフィック，サウンド関係の華々しい記事が目立つようになる。4月号からの表紙はもとのりゆき氏にお願いした。

81 1月号



82 2月号



83 3月号



ぶっとんだゲーム特集が衝撃的！

84 4月号



⑧5月号



⑧6月号



⑧7月号



⑧8月号





⑧9月号



⑨10月号



⑨11月号



⑨12月号



4月号から表紙デザインを一新。須藤牧人，塚田哲也両氏のCGが交互に本誌を飾るようになった。さて，'90年代のOh!Xは，などと能書きを垂れている暇はない。時代はリアルタイムに動いている。Oh!Xはどこへ行くのか？

⑨1月号



⑨2月号



⑨3月号



⑨4月号



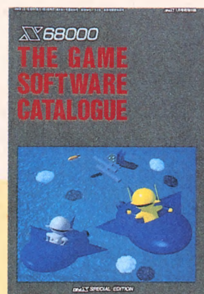
ついにディスクが付録

⑨6月号



⑨5月号





1月号付録のX68000ゲームソフトウェアカタログ

⑨7月号



⑩8月号





6月号付録の創刊8周年記念PRO-68K

お祝いの言葉

へーえ，100号？　そうか，まだ100冊しか出てなかったのか，もっといっていると思ってた。まあ，100冊ったって，数字なんてどーでもいーことさ。さるお方の結婚式ももうすんだし。過去も未来も似たようなもの。大事なのはその100冊に散らばる過去の名作たちだ。逆立ちしてもOh!X(Oh!MZ) でしか読めない，機種の壁を越えた名作・奇作・珍作の嵐。これが財産である。Oh!X傑作集を出したいくらいだ。

いま，その個性も矢面に立たされている。浸透は常に拡散を伴うからだ。いくつものベクトルを内包した新しいスタイルも必要とされるだろう。しかし，知識より知恵，実用より心，完成されたプログラムよりマシンポテンシャルの開拓精神の基本は変わらない。X68000はまだまだ深いポテンシャルを秘めている。のんびりしている暇はない。そして粋なパソコン誌として，多様化する読者と共に，Oh!Xは100万部を目指すのである。

からころも　きつつなれにし　つましあれば
はるばるきぬる　たびをしぞおもふ
てなもんだ。めでたいな。　（荻窪圭）







SOFTWARE INFORMATION

今月は夏休みに向けてか，ひさびさに大量の新作の情報が入ってきました。てなわけで，今回は4ページでお届けすることにします。しっかし，毎月コンスタントにこのくらい発表されればありがたいのに……。

話題のソフトウェア

いや〜，先月は梅雨だなんて書いてしまったもんだから，皆さんからのお叱りのハガキの多かったこと。まあ6月18日を予想して書いているんだから，そーゆーこともたまにはあるわな。許せ許せ，ハハハ。というわけで，今月こそ梅雨です。じつにうっとうしいですね〜（え？　フォローになってないって？　でも，梅雨明けって7月22日って気象庁が言ってるからいいじゃない）。そういや，もうじき夏休みですねぇ。クーラーの効いた涼しい部屋でアイスティでも飲みながら，ゆったりとゲームに浸る。う〜ん，極楽極楽（とか言ってすっかり違う方向へ話を持っていくヤツ）。悪いことは全部忘れて，夏休みの前半は遊びまくりましょ。宿題そのほかで青くなるのは，来月号が出てからでも十分なんだから……（ホントか，おい）。

さて，夏休みを目前に控えて，ゲームのほうもバタバタと活気を増してきました。なんともうれしいぢゃあ〜りませんか。うれしさ爆発，ページも倍。これを書く側としては，ほんっとに喜んでいいやら悲しんでいいやら……。ま，そんなこと言っててもしょうがないんで，順を追って紹介していくことにしましょう。

まずはこの**ギャラガ'88**。電波新聞社よりすでに発売されているので，もうクリアしちゃった人もいるんじゃないかな。このゲーム，3年ほど前にゲーセンで流行ったナムコのシューティングなんだけど，たった3年前なのに第一印象で“懐かしい！”と感じてしまいました。もっとも私の場合はこのゲームの元祖，ギャラクシアン（死語だよなあ）を中学生の分際ながら（あん，年がバレる）ゲーセンで遊んでたから，そのとき印象が強いからかもね。で，肝心の出来ですが，これがなかなか。プーッとふくれるハエさんや，かわいいボーナスステージのギャラクティックダンシングもゲーセン版同様いい味出してます。さすがに先に移植されていたPCエンジンよりは，グラフィックもきれいですし。これはゲーム自体は，そう難易度の高いシューティン



がんばで，ぐただだ！

		(前回順位)
1	ポピュラス	1
2	グラナダ	4↑
3	ワンダラーズ・フロム・イース	3
4	ダンジョンマスター	2↓
5	天下統一	－初
6	スーパーハングオン	－↑
7	ジェノサイド	10↑
8	三国志II	5↓
9	サーク	6↓
10	ソーサリアン	7↓

疲れたー。いつもはサンプリング抽出をしているのに，今月は28日までのハガキを全部カウントすることになってしまいました。手伝ってくれたみんな，ありがとね。

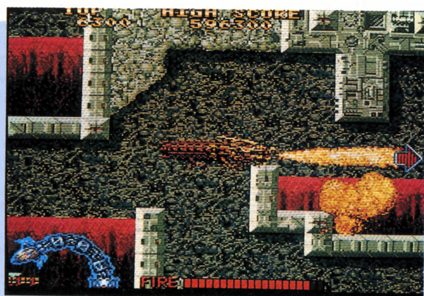
さて，100号記念（かどうか知らんが）の完全集計版TOP10。ランクアップ・ダウンもつけてみたけどどうでしょう。

おやおや。そろそろみんな解き終わったと思ったらダンジョンマスターは4位まで落ちてしまったぞ。みんな結構ドライだな。代わって2位の座を手に入れたのは，グラナダ。これはウルフ・チーム最高順位！　イースファンのみなさん，もう少しだったのに，残念でしたね。

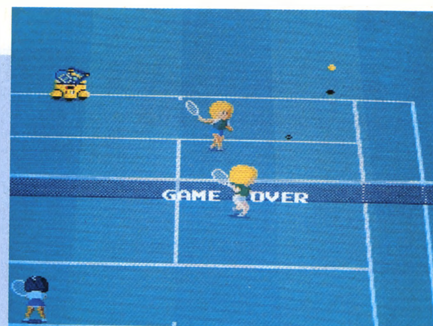
そして，5位初登場天下統一。このゲームの



パズニック



サイバリオン



ワールドコート

グでもなかったので，ゲーセン版のほかにも，X68000用にオリジナルステージも用意されています。こちらもぜひプレイしてみてほしいですね。

さて発売中といえば，ブロダーバンドジャパンの**パズニック**。こちらもゲーセン版（タイトー）からの移植です。ゲーセンではじっくり考えているヒマがなかったので，かなりお金を注ぎ込んだ人もいることでしょう。同じマークのブロックを隣接させて消していくパズルゲームなんですが，ブロックは重力の関係で上にあげられないし，でもってタイミングが命の面もたくさんあるしで，一筋縄ではいかず悩むわけなんです，これが。家でじっくり楽しめるようになれば，クリアも夢じゃなくなるかな。でもムリかな，私バカだから。

でもって，同じパズルゲームであるコナミの**クォース**ももう発売されていますね。こちらもゲーセン版からの移植もの。ゲームボーイなどでも発売されているし，けっこうやり込んでいる人もゴロゴロいるのでは？　このゲームはシューティングの要素も含まれているので，ちょっとだけ反射神経が必要かもしんないけど……。

ん？　こうやって書いていくと，なんかゲーセン版からの移植ものばっかりだわねー。ま，いっか。ついでだから，このまま続けて移植ものを一気に書いてっちゃおうっと。

じゃ，次，**サイバリオン**。このゲームはドラゴンを操って，矢印の指し示す方向へ進んでいくタイトーのアクションゲームなんだけど，ゲーセン版はスティックじゃなく，トラックボールでってところがミソだったよね。今回はジョイスティックでもできるようになっているけど，通ならやっぱりトラックボールで遊んでほしいな。ジョイスティックに慣れているからこそ，トラックボールで遊ぶっていう感覚は新しくっていいかもしんないし。8月中旬にシャープから発売される予定。いま頑張ってSPSさんが移植しているので，楽しみにしてて。

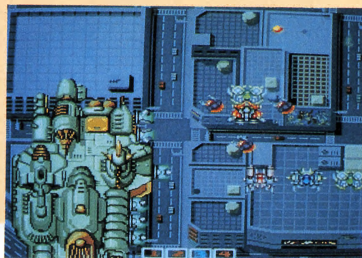
でもって，同じくSPSさんの移植によるナムコの**ワールドコート**の登場です。このゲームってば，地味なスポーツゲームと思いきや，結構ハマりやすいゲームだったりするわけ。その当時は友達同士で遊んでいる高校生や予備校生をよく見掛けました。そうこうする間に，PCエンジンにも移植されちゃったりなんかもしたし。さすがに今回はクエストモードはないみたいだけどね。スマッシュやサーブがうまく決まるようになると，もうまさにテニスの選手になった気分で楽しめます。そういや，わざと女の子の選手を転ばせてパンチラを楽しんでいたふとどきものもいたっけかなー。まあ，それはおいといて，このゲームは7月20日に発売される予定ですのでお楽しみに。

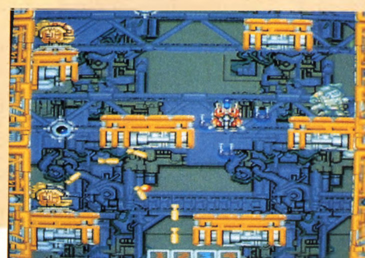
さてお次は，じゃ～ん，**イメージファイト**なんですねー。このゲームはかなりムズかったんで，わりとマニア受けしていたシューティングです。アイレムさんのゲームはあのR-TYPE以来だから，このイメージファイトの登場を待ち望んでいたユーザーも結構いるはず。その夢がやっと実現しました。このゲーム，ポッドと呼ばれるアイテムを，いかにうまく使いこなすかがカギとも言えるでしょう。これをうまく扱えないと，かなり苦しい。はじめてやると全9面クリアどころか，5ステージクリア後にある補習ステージにたどりつくのにもてこずったりするんですよ，これが。で，移植の出来はというと，画面写真を見てのとおり。なかなかよさそうでしょ？　コンティニューもあるらしいから，ゲーセン版では見ることができなかったエンディングも見られるかもしれないぞ。年内発売の予定だから，詳しい情報はもうちょっとだけ待っていてね。

イメージファイト
これまたゲーセンで人気だった超ムズいシューティングゲーム。なかにはゲーセンで血を流した人もいるとかいないとか……。









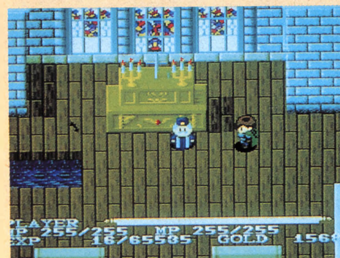
評判は……あれ，ハガキはAFTER REVIEWに行っちゃったの？　じゃあすいません，そっちを見てちょうだい。

その下に謎のカムバック，スーパーハングオン。確かに長く遊べるが，なぜ今になって……。さらに7位ジェノサイドのランクアップも謎だ。もうすぐラグーンも発売されるというのに……。そういや，みんなCDはもう買ったかな。

あやや，三国志IIもソーサリアン（まだいる！）もランクダウンか。先月威張ったのが反感を買ったかな？　こりゃおいらは静かにしてたほうが良さそう。……（それじゃ，また来月）。

（浦）







ラグーン
ジェノサイドで人気のソフトハウス，ズームの期待の第２作。今度はアクションRPGだぞ。２頭身のキャラクターがなんとも可愛らしい。期待度大のゲームだ。



実戦ビリヤード

まあ，ゲーセンからの移植情報はこんなもんかな。もうちょっとすると，またいくつか出てくるみたいだけど，それはそれでまたあとのお楽しみということで，ね。

じゃあ，今度はゲーセンものではないやつをガシガシ紹介していくことにしましょうか。

まずは，皆さんお待ちかねのズームの**ラグーン**からいきましょう。ジェノサイドで一躍人気者となったズーム。その第２弾といえば，アクションゲームファンでなくとも気になるところ。開発状況はわりとよいようで，発売に向けて着々と進行している様子です。今回は，最終段階に入ったともいえる現時点での画面写真をお届けしましょう。ジェノサイドであれだけ頑張ってくれたズームが，アクションRPGという新境地でどういった展開を見せてくれるか，楽しみにしたいですね。

さて，バトルチェスでX68000に参入したパック・イン・ビデオからは，**実戦ビリヤード**が発売中。このゲームは，その名のとおりビリヤードゲームで，ナインボールやローテーション，はたまた４つ玉（知ってるかな？）までプレイできちゃいます。プールバーなるものが乱立したビリヤードブームはもう過ぎてしまいましたが，本来ビリヤードというものはじっくり玉筋を読んで楽しむものだし，家でゆっくりビール片手にパソコンに向かって楽しむのもいいんではないでしょうか。

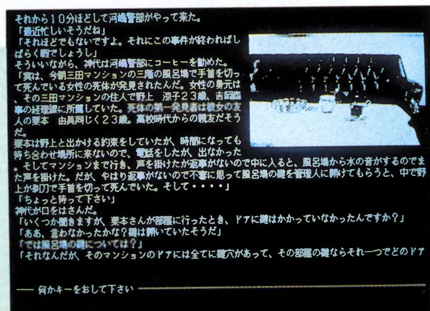
じっくり楽しむといえばやっぱり**Misty4**でしょうか。一連のMistyシリーズの第４弾です。前作からしばらく間が空きましたが，やっぱりデータウエストさん，頑張ってくれました。今回もユーザーからのシナリオ５つを中心に構成されてます。暑い夏に，ちょっとサスペンスタッチの推理ゲームを静かに楽しむ，なんて大人っぽいじゃない。ところでデータウエストといえば，ピンとくるのが第４のユニットシリーズ。ブロンウィンファンの皆さん，ご安心を。シリーズ第５弾**D-Again**も着々と進行している様子。今月はまだ画面写真をお届けできないけど，もうちょっとしたら詳しいことをお伝えできそう。待っててね。

でもってT&Eからは**ルーンワース～黒衣の貴公子～**が発売，ドラマチックな展開で進んでいくアクションRPGです。なぜドラマチックかというと，このゲームはプレイヤーの行動によって，たどるストーリーが変わっていくからなんですねー。いわば，あなたがストーリーを作り上げていくゲームなのです。うん，これは奥が深いぞ。

またT&Eでは次回作**幻獣鬼**を開発中。これはサンプル版をプレイしたところによると，敵の攻撃が，というか敵の放つ弾が雨アラレのごとく飛んでくるので，なかなかタイヘン。やりがいがありのようです。そのほか，あのゴルフゲーム**遙かなるオーガスタ**も出す予定だそうだし，今後のT&Eの動向には目が離せない!?

さてさて，数々のラインアップを控えているザイン・ソフトでは，ただいま**REINFORCER**と**バルーサの復讐**をしゃかりきになって開発中のよう。REINFORCERのほうは，トップビュータイプの８方向スクロールという，サイバーパンクアクションゲームだそう。こちらは先月号でも紹介しましたが，さらに開発が進んだものが手に入ったので紹介しちゃいましょう。発売は９月上旬の予定。一方のバルーサの復讐のほうは，剣と魔法で攻撃するファンタジーアクションゲーム。サイドビュータイプで，８方向多重スクロールするというシロモノ。こちらは７月発売を目指して，目下頑張って開発中とのこと。お楽しみに。

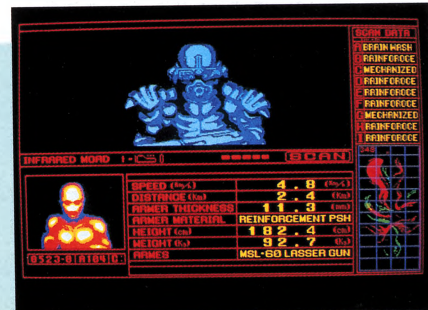
あっ，とついうっかり忘れそうになっちゃった，いまや読者の人気ナンバー１に輝いたポピュラス。そのポピュラスの追加シナリオが発売になったことは，きっともう皆さん周知の事実でしょう。今号のREVIEWでも紹介していますしね。まあ，それはおいといて，なんとそのポピュラスを発売したイマジニアから，**シムシティー**が移植，発売されることが正式に決定しました。



Misty4

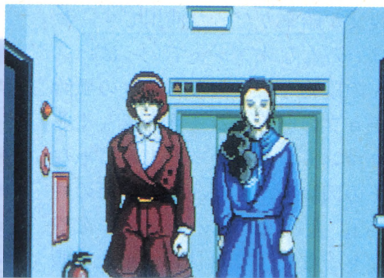


幻獣鬼



REINFORCER







闇の血族
サコムのノベルウェアシリーズ。推理探偵もので主役はうら若き乙女。リアルな感じのグラフィックが雰囲気を出しているよね。

わーい，パチパチパチ。このシムシティー，都市開発を題材にしたリアルタイムシミュレーションで，14個のアイコンを駆使して町を発展させることが目的。鉄道を敷いたり工場を建てたりとなんとも忙しい。まあ，詳しいことはまた来月にでも紹介させていただきますのであしからず。へへっ，出し惜しみしちゃってごめんね。また，イマジニアではポピュラスの原作者であるピーター・モリニュー氏の来日を記念して，ポピュラス大会を企画しています。我こそは，と思うポピュラスマニアの方，んあ？と思ったらプロミストランドのREVIEWの左下を見て，応募してください。よろしくね。

さて，移植といえばスタークラフトの**トンネルズ＆トロールズ**。こちらもすでに発売になりましたね。もともとテーブルトークRPGということで，そのあたりが好きな方には熱狂的な支持を受けているゲームですが，ようやっとX68000にも登場。ほっとした方もいることでしょう。このゲームは，背景となる舞台設定がしっかりしているので，はじめてRPGをやる人でも親しみやすいかな。それにオマケとしてオリジナルオーナーズカードや，ドラゴン大陸のポスターなど，RPG必携3点セット（！）なるものが付いてくるなぞ，ニクい心配りがうれしいじゃありませんか。毎日コツコツとたゆまぬ努力をしても苦にならない方は，ぜひプレイしてみては？

あちらものの移植じゃあないけれど，こちらも移植もの。PC−9801からの移植だけれど，システムソフトから**遊撃王II**がでるそうです。PC−9801版ではサイバースティックが使えるってんでびっくらこいたわけですが，当然のことながらこのX68000版でもサイバースティックが使えます。フライトシミュレータゲームなので，サイバースティックを使えば，パイロット気分で楽しめそう。画面写真もお届けできなかったし，発売はまだ未定だけれど，出来はかなりよさそうですよ。期待度大です。

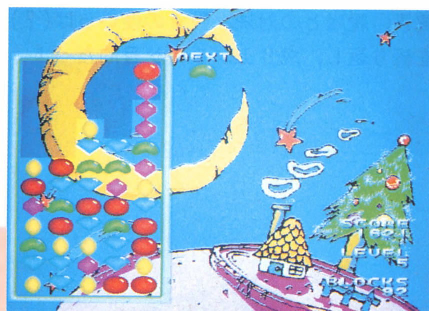
さてと，そいじゃシステムサコムだ。**ジェミニウイング**の開発も佳境に入ったカンジなのだけれど，その一方であのノベルウェアシリーズである**闇の血族**の開発も，しっかり進行している様子。今回は女の子が主役のアドベンチャーとあってか，サコムとしても主人公のグラフィックにはリキを入れているよう。届いたばかりのグラフィックの数々を紹介しますね。この闇の血族は，7月か8月には発売されるそうなので，ノベルウェアファンは見逃せませんね。

さて，最後を飾るのはM.N.M.Softwareです。今回紹介するのはThriceとPipyan。ThriceはColumnsタイプのパズルゲームで，縦，横，ナナメに同じキャラクタを3つ以上揃えて消していき，得点を競うというもの。なんと300位までネームエントリーができるそうな。ふえ〜。でもって，このタイプはずーっと画面を見ているだけでは疲れてきちゃうこともあるので，それをなくすためにもある点数をクリアするごとに，背景がいろいろと変わっていくので，飽きずにプレイできます。8〜9月に発売されるそうなので，Columnsにはまった人はぜひプレイしてみてください。そしてPipyanは，倉庫番のように男の子のキャラクターを操作して，ブロックをうまく組み立てていくといったゲームです。さながら工事現場のようなステージ上で，あたふたと動き回る男の子，失敗するとペコペコと頭をさげたりなんかして，とってもキュート。こちらは，7月中旬にタケルより発売される予定とのこと。ひょっとしたらこの本が出る頃には発売されてるかもね。

てな感じで今月もそろそろネタ切れです。こうやってずらっと書いたあとで見てみると，おや，X1がひとつもない？　んなバカな！　でもほんと，そうみたい……。なんかとっても悲しいなぁ。ああ，X1ユーザーの怒りの声が聞こえてきそう。ではまた来月。



トンネルズ＆トロールズ



Thrice



バルーサの復讐



Pipyan

●大航海時代



ロマンたっぷり 大海原で帆船の冒険

Urakawa Hiroyuki
浦川 博之

「維新の嵐」に続く光栄のRÉKOEITION GAME第2弾は，中世の帆船の旅をシミュレーションゲームにした「大航海時代」です。貿易，艦隊との対決，数々の使いっぱ(?)を繰り返し，成り上がるのが目的だっ！



X1turbo用 5"2D版 4枚組 9,800円(税別)
光栄 ☎044(61)6861

ども。親父が船乗りだった浦川です。おかげで家はいろんなオブジェでいっぱい。おさるさんの置物とか巨大な素焼きの風鈴とか，ダチョウの卵とか。こう節操なく並ぶと海のロマンもなにもあったもんじゃない。で，その因縁か，私が光栄の「海のロマンゲーム」，大航海時代のレビューをやることになりました。これは1500年代初頭を舞台とした海洋シミュレーションです。ポルトガル，イスパニア，イスラムによる貿易の主導権争いの真っ只中の頃ですね。プレイヤーは有象無象の商船長の中のひとりとなり，地中海に始まって，アフリカ喜望峰，アラビア，インド，はてはジパングまで航路を開拓し，貿易を行います。

貿易のほかにもうひとつ，貴族の爵位を得るというフィーチャーがあります。主人公の先祖が航海の失敗から爵位を剥奪されたという設定になっていて，お家の復興がプレイヤーの悲願なのです。オーイエー(面白度1)。ライバル国の艦隊をやっつけたり，勅命を遂行したりして国王に認めてもらい，最高爵位まで昇りつめるべくこれまた世界を駆け巡るわけですな。

ややこしそうに聞こえるかもしれませんが，「貿易する“スタークルーザー”」といえばわかるかな(もしくは光栄版“WARNING”か?)。

地中海の隣人

私はタ＝バスコ＝ガマ。ちょいと辛口のいい男。自分ではちょっとだけ銀英伝のラインハルトに似てると思っている。親父が遭難して行方不明になったので，家の再興のために大海原に出て一旗上げることにした。といっても，手元にあるのは親父の残した小さい商船だけ。最初はヨーロッパ周辺で経験を養い，財力をつけねばならない。幸い，頼りになる昔の父の部下ロッコがいる。ひとりでもロッコとはこれいかに?

ロッコ「ぼっちゃん，禅問答してないでこれからどうするか決めてくださいよ」

じゃあ酒場に行こう。情勢も知らずに積み荷を仕入れちゃ失敗は目に見えてる。

カランコローン。

Yo「あら，いらっしゃい。」

ようこちゃん，ここのみんなにwellsスーパーマラソンね。

ロッコ「おや，誰か来やすぜ」

男「あんた，リスボンで何か仕入れるんだったら，砂糖を買うといいぜ」

かくして1502年2月，タバスコ一行と砂糖をどっさり載せたラテン船「難破1号」は大西洋へ漕ぎ出した。……誰だ，こんな不吉な名前つけたやつあ。

航海中の画面は下の写真のとおり。1画面が緯度・経度ともに約5度の広さだ。この左側の矢印はなんだろう。

ロッコ「上は針路。真ん中は風力計でさあ。左上の数字が風力で，その下は潮流計」

いまは逆風だな。三角帆だから逆風でもわりと速いんだよな。速い速い……(ゆるゆるゆる)，速い……。おい，遅いぞ。なんだこの遅さは。おまけに夜が明けるたびにディスクはガーガー鳴るし。

ロッコ「この辺りは外洋と違って風がおとなしいですからね。それに海を航行してるのはわしらだけじゃねえんすから，処理速度もちったあ遅くなりまさあ」

ぶーぶーいいながら，3日でイスパニアの首都，セビリアに到着。幸い，砂糖は約2倍の値段で売れた。元が安いからあまり大きな儲けにはならないが，楽な航海だったからこんなもんだろう。しかし，どこの港も人の顔が全部一緒だな。旅情ってもんがない。酒場の娘の顔は違うんだけど。

ロッコ「なにぶつぶついってんです。次はどこへ向かいやすか?」

神聖ローマ帝国のピサで美術品が安く買えるようだから行ってみよう。

再びゆるゆると地中海を進む。このゲーム，舵を切るときはメニューを開かなくてはならない。そのたびにディスクアクセスするので，地中海のような入り組んだところを航行するのはなかなか骨が折れる。

十数日の航海を経て，ピサに到着。すいませーん，美術品くださーい。

交易所の親父「美術品は金貨310枚だよ。いくつ買うかね?」

買えるだけ全部。ところで，この美術品って中身はなんなの?

親父「見てみるかい(ごそごそ)。ほら，



こっと

名物“ピサの斜塔ぶんちん”。いまなら大小の鉄球もつけちゃう」
ガリレオの実験は100年後なんですが……。

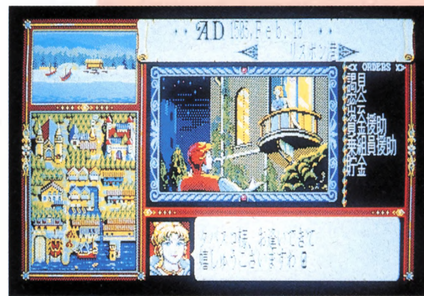
ザ・グレーテスト・ミッション

半年近く地中海を駆け巡ったおかげでめでたく2隻目の船を購入できた。名前はもちろん“難破2号”。途中酒場で知り合ったオスワルドという男に船長をまかせる。
地中海の主な貿易ルートは次のとおりだ。
・リスボン（砂糖）→セビリア
・アントワープ（陶磁器）⟷ロンドン(羊毛)
・ピサ（美術品）⟷マジョルカ（穀物）
もっとも，港ごとに物価は違うし，ほかの艦隊の取引によっても相場は変動するので，絶対これというパターンはない。それから「イスタンブールの美術品はいいぞぉ，儲かるぞぉ」とさんざん吹きこまれたが，ポルトガルとイスラムの仲が悪いので立ち寄っても追い返されてしまった。王様，なんとかしてよ。トホホ。
さて，そんなある日。立ち寄った酒場で見知らぬ男に呼び止められた。
男「よう，あんた。タバスコさんだろ。マジョルカであんたを捜してる奴がいたな」
ロッコ「なんでしょうね，ぼっちゃん？」
デ，デートの申し込みかな?（ずで）
耳を引っぱって連れていかれたマジョルカ港では交易所の親父が待っていた。
親父「わざわざどうも。あなたに頼みたいことがあって捜していたんです。実は陶磁器で儲けようと思うんですが，35ほど仕入れてきてもらいたいんです。金貨4620枚で仕入れてきてもらえますか？」
わざわざ呼びつけて使いっぱかよー。
ロッコ「そういうことといってちゃいけやせん。かなりワリのいい仕事なんすから。それに交易所御用達になれば王様のお目に止まる日も近いですよ」
ぶーぶーいいながら申し出を受け，ヴェネチアで陶磁器を仕入れてくる。さっさと引き渡し，その報酬で飲んでいると……。
男「おい，タバスコさんだろ。リスボンで王様がお呼びだっていう話だぜ」
ロッコ「やりやしたね，ぼっちゃん！　すぐに駆けつけやしょう」
もちろんだぁ。この家名復興のチャンスを逃がしてたまるか。リスボンに急行だ！
ゆるゆるゆる。リスボンを目指して帆船はのんきに進む。リスボンに着くや否や，一目散に城へ駆けこんだ。
役人「謁見の申し込みか？　しばらく待たれよ。……。陛下がお会いになるそうです」
荘厳な謁見の間に通される。国王が現れた。面を上げる。緊張の一瞬。
ポルトガル国王「おお，そなたがタバスコか。お前を呼んだのはほかでもない。実は羊毛が38必要なのじゃが，そなたに……おいおい，どうしたのじゃ？」

タバスコ,南へ

勅命の使いっぱを完遂した私は子爵の称号を賜わった。あれからイギリス，北欧まで足をのばし，貿易網はイスラムを除いたヨーロッパを網羅している。新たに中型の船を購入して旗艦とし，ポチョムキン号と名をつけた。
さて，ロッコ，新しい船も手に入ったし，ここらでひとつアフリカに行ってみようと思うのだが。あそこじゃ金が手に入るという話じゃないか。
ロッコ「うーん，ちょっと装備が弱い気もしやすが，いつまでもヨーロッパでもないですしねぇ」
よし，決まりだ。食料と水を満載し，ひたすら南を目指す。セビリアから2,3日ほど行くと海の色も変わり，アフリカに入ったことがわかった。ちなみにBGMも変わる。おお，風が強くなってきたぞ。わあ，強い強い。風力8だ。暴風だぞ，こりゃあ。
ロッコ「これが外洋の風でさあ。これに乗って一気に南下しやすぜ」
てててててて。信じられないペースで船は進んでいく。うわあ，揺れる揺れる。きぼぢわるい，げろげろ。ちょっとアフリカは早すぎたかなあという思いが頭をよぎる。
ロッコ「ぼっちゃん，港が見えやす」
え？　もう着いたの？　まだ1週間そこそこなのに。しかし，交易所には金がいっぱい！　有り金はたいて全部買い込む。はっはっは。帰れば大金持ちだぞ，ロッコ。
てててててて。帰りも快調。見事アフリカ金貿易航路が開けたかと思われたが……。
「提督。嵐だ！」，ざざーっ。もりもりと海が盛り上がり，船はひっかきまわされた。「舵がききやせんぜ！」。西を向きながら，船は東へ押し流される。もうムチャクチャ。「難破1号の姿が見えやせん！」
海は一昼夜荒れ狂い，さらに難破2号までが行方不明になった。やはり名前が悪かったか。旗艦ポチョムキンも食料の半分と3分の2近い乗組員を失った。安易に外洋に来るんじゃなかった……。と，放心状態でさまよっていたのも束の間。
「提督。嵐だ！」
この船の末路が私の脳裏をよぎった。


酒場は大事な情報源，そのほかにもいろいろ……


王女クリスとの密会，たまにはこういうのもね



戦闘画面はやっぱりヘックス

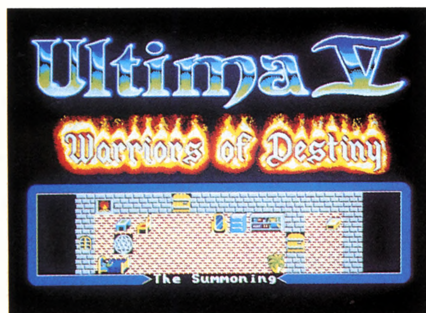
〈ちょっとひとこと〉

貿易が題材のゲームというのは，どうしても単調になりがちです。イベントなどを設定してうまく防いではいますが，操作性の問題が目につきやすい序盤では「ずーっとこんなことが続くのか」と目まいを覚えてしまうこともあります。
地中海を出るようになれば，自分で航路を開く楽しみもあって，自分の好きなように遊ぶことが可能になります。規制が緩く，自分の好きなように遊べるのが身上です。なんだかんだいって結局ハマってしまうのが光栄のすごいところ。
最後にBGMのことですが，音楽性がないとはいいませんが，「3パートしか使わないBGMを聞かせてCDを売り込むのはちょっと無理があるんでないの」ということは指摘しておきましょう。

冒険心刺激度	10
マニュアル親切度	9
グラフィック	8
操作性	6
BGM	4
熱中度	8

THE SOFTOUCH

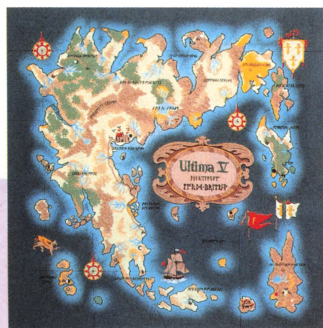
●ウルティマV



天下無敵のシリーズ第5弾

Ogikubo Kei
荻窪　圭

その面白さがわかる人にはすごく面白い。そういう一風変わった，しかも奥深い魅力をもつウルティマシリーズの5作目がいよいよ登場。さらなるリアルさと難解な謎であなたの頭を悩ませる？



X68000用　5"2HD版2枚組　9,800円(税別)
ポニーキャニオン　☎03(221)3161

ああ，ダンジョンマスターって，なんて楽なゲームだったんだろう。メモを取る必要はほとんどなかったし，地下6階までは下を目指して進んでいけばよかった。

ウルティマはそんなわけにはいかない。右も左もわからない大陸の真ん中に放り出され，行くも地獄行かぬも地獄，森の木陰でドンジャラホイ，なのである。世界の合言葉は森ってなもんだ。

どーして怠慢で出不精で睡魔に魅入られた私がウルティマVなどという超大河，スーパー大河なゲームをすることになったのかというと，ウルティマIV経験者がほかにいなかったからである。経験者っていうだけで終わらせたわけではなく，しかも3年前，友達の部屋のX1turboIIで遊んだものだったりするので，当然育てたキャラクターは持ってこれないし，当時集めた膨大なメモは引っ越しの際にみんな捨てちゃったしの後悔先に立たず，あとの血祭り村祭り，かんなん汝を玉にするってな状況。人生，蜜のように甘くタバスコのように辛し。

懐かしい風景,旧知の友

イオロ，シャミノ。記憶の底にこびりついた青春の残滓から消え去る寸前のデータベースにこびりついていた懐かしい名前。こんなことまで覚えているなんて。いや，覚えているというより思い出すことができるといったほうが正しい。あくまでも画面にその名が記されたとき，懐かしさを感じるだけだ。役に立たない記憶。

主人公はアバタール。アバタールというのはAVATAR,アバターとかアヴァターラなどともいう。「化身」とか「権化」という意味である。化身といえばレインボーマン。レインボーマンは月の化身，火の化身など7種類の化身になれた。つまり，アバタールだったわけである。レインボーマンといえば「インドの山奥で修行」。このインドがポイントでアバタールというのはもともとインドの言葉だったのだ。インドにおいてヒンズー教のヴィシュヌ神は人々の前にさまざまな動物や人の姿を借りて現れると考えられ，それを化身（民衆を救おうとして神が姿を変えて現れること，あるいはその姿），つまりアヴァターラと呼ぶのだ。

で，ウルティマVの主人公はウルティマIVで8つの徳をすべて極め，アバタールとなった者なのである。私はなった覚えがないがなったらしいのである。なった覚えがある人(つまりウルティマIVからキャラクターを移した人）は，それなりのレベルから始められるが，私のようにアバタールになった覚えのない人はアバタールのくせにレベル2という苦難の始まりとなる。弱い弱い。

舞台はウルティマIVと同じ広大な大陸だ。しかし，前作でとったメモがない。最初からやりなおし。それでも歩いているとだんだんと思い出してくる。ここに村があった，この辺にムーンゲートが出るはずだと。

自由の持つ厳しさ

ウルティマがほかのRPGと異なる点はゲームを進めるためのガイドがまったくないことである。イースを代表とする日本式RPGはスゴロク型であった。ダンジョンタイプのRPGも，その存在自体にダンジョンを深いところへ向かって降りていくという不文律的ガイドがあった。しかし，ウルティマは恐ろしい。前向きRPGではなく，はなから，大陸の真ん中で右往左往，どこから手をつけてどこへ向かうのかも自由なのだ。かなりレベルが上がった後半にならなければ行けないような場所でも，然るべき情報と金を出して買えるアイテム（船など）があれば行けてしまうのだ（ちなみに，キーバッファはたまらないぞ)。

つまり，ドラクエやらイースやらのスゴロク型RPGが管理された，安全だけど自由のない日本であれば，ウルティマは自由だけど危険ですべて自分の集めた情報を基に自分の判断で動かねばならないアメリカなのだ！　ほほほほほぅ。あなたはどっちが好きですか。自由社会？　でも，自由の旗のもとで自由に生きていくためのプレッシャーは相当なものである。

たとえば，ウルティマではお城のオークの樽に隠されたアイテムを盗むことも，寝ている衛兵を殺すことも簡単だ。本当に簡単だ。しかし，その結果がどうなろうと自分の責任である。特に，ウルティマVは平和で善良な人々ばかりであったIVと違って



やったー，ついに亡霊登場でレベルアップだ

邪悪なブラックソーンの支配下にあるのだ。その中でアバタールとしての行動をやり通さねばならない。不当な要求に答えて「持っている金の半分を衛兵に支払う」のも，信念を貫いて「牢獄にぶちこまれる」のも自由だ。

うーん。このゲームは「うんちゃらうんちゃらの自由」を要求するガキの精神に「自由の持つ厳しさ」を叩き込む教育ゲームだったのか。私はもちろん，血反吐を吐きながらでも，管理された健全な社会よりアナーキーで自由な社会のほうを選ぶ。日本という平和で安全な社会が好きな人はガイドに沿って大陸を旅するドラクエでもやっていてください。

複雑怪奇な社会

ウルティマⅤには表の世界と裏の世界がある。表の世界がブラックソーンに支配された圧政の社会であり，裏の世界はロード・ブリティッシュに忠実な人々が集まった，レジスタンスである。レジスタンス，そんなものまであるのだ。アバタールである主人公とウルティマⅣでともに戦った仲間たち。もちろん，レジスタンスとともに行方不明になったロード・ブリティッシュを捜し，この世界に平和と徳を取り戻すのだ。それが目的だ。それにはアバタールはアバタールらしく行動せねばならない。ものを盗むな，罪のない人は殺すな，邪悪な者に対しては勇敢であれ。

何が自由だ！　道徳的であらねばいけないなんて！　規範だらけではないか。しかも目の前にはおいしい餌がぶら下がっているというのに，道徳的であるために自らを律せねばならないのだ。目の前の快楽に弱い荻窪圭はどーしたらいいのだ。

昼と夜

話はがらっと変わる。ウルティマⅤのうりのひとつに，時間がある。街の住人は朝になると起き，働き，昼になると食事をし，夜になると寝る。だから，買い物をしようと思ったら店が開いている時間に行かないと売ってくれない。夜になると門を閉められて入れない街もある。みな働き者で規則正しい生活を送っているのだ。なんと，夜になると会合を開いているレジスタンスの農民もいる。門番の衛兵もちゃんと食事どきや交代時間には入れ替わる。ベッドももちろん住民の数だけある。私は宿屋のない街では他人の家の他人のベッドで休ませてもらう。こんなリアルな街にも「不法侵入罪」はないみたいで，誰も咎めない（これはたんなる皮肉）。



あまり自由を満喫しすぎるとこういう目にあう



しゃべる馬の「エド」，じゃなくて「スミス」

おおむね，圧政者がいても住民は善良である。が，しかし，巡回する邪悪なシャドーロードがいる。シャドーロードがいる都市に入ると憎しみの空気や臆病の気配を感じるので，そんなときはその都市はやりすごすのがいい。シャドーロードがいる都市の衛兵は私らを見かけると有無をいわさず逮捕し，商人は金をちょろまかし，住人は会話しがてら何かを盗む。シャドーロードに捕まったら大変で，まず勝てない。しかし，悪いのは衛兵や住人ではないので，怒ってはいけない。

最後に，ウルティマⅤで遊ぶのに必要なものを書いておこう。

ひとつは根気である。なにせ，スーパー大河であるから。レベルアップも経験値をためるだけではだめで，ロード・ブリティッシュに会わねばならないのはウルティマⅣと同じ。ただし，Ⅴではロード・ブリティッシュは行方不明なのだ。そっと教えるとキャンプ中に亡霊が現れてレベルを上げてくれることがあるのだ。うーん，根気の野外キャンプである。

続いて異種世界，異種文化を楽しむ心である。優れたファンタジーはリアルな異文化を持った世界が描かれている。読者はその異文化を楽しむのである。劣ったファンタジーは現実世界をひきずった文化の上に成り立っているため，想像力をあまり要求されず読みやすいが，ファンタジーとしての魅力に欠ける。

さらに，筆記用具である。いつ，どこで役に立つかわからない膨大な情報。あっちへいったりこっちへいったり。メモが必要だ。経験を語ろう。ユーの街から別の都市へでかけると，ユーの街の誰それが知ってるよといわれた。すぐにでも欲しい情報だったのでユーの街へ戻って尋ねた。すると，君は俺がそれを知っているということを誰から聞いたんだい？　といわれた。そんなことまでメモしてなかったので，また危険な森を抜けて戻り，名前を確認し，またユーの街へ戻った。メモは重要。

それでもって，英和辞典である。なんといっても英語だ。たとえば，立て札や墓碑銘，看板にはルーン文字で書いてあるものがたくさんあるのだ。そして，それを表に従って解読すると英文が現れる。それを訳さねば何が書いてあるかわからないのだ。ほかにも英語がわかったほうがよい場面はある。このルーン文字を訳すのが面倒なこととことこと。うーん。

では，みなさん，頑張ってください。ウルティマⅣをやっていない人でも，終わってない人でも大丈夫です。

総評だべさ

良くも悪くも，伝統と格式に守られた底の深さと指10本を駆使する操作性はウルティマである。誰の文句も許さない強さだ。ほとんどローリングストーンズのようなものだ。スターウォーズのようなものだ。

世の中にはちょっと聞いた分には耳に優しくてノリやすくてヒットする歌謡曲や売れ線ロックと，ちょっと聞いただけでは異質で馴染めないけれど聴き込むほどに味の出る名作がある。ウルティマは後者のほうだ。ウルティマワールドに馴染むほど，味が出て，面倒だなんだと文句をいいながらついつい大陸をさまよったり会話にうつつを抜かしてしまう。ストーンヘンジ4000年の歴史というか，ケルト人3000年の歴史というか，孔子の儒教2500年の歴史というか，ディズニーランド35年の歴史というようなそんな重みは重いのである。

5段階評価

ウルティマ度：★★★★★
ロード・ブリティッシュ度：★★★★★
非ドラクエ度：★
非イース度：★★★
道化師殺人事件度：★★

＊

アメリカンジャーニー度：★★★★★
カリブの海賊度：★★★
ジャングルクルーズ度：★★★★
シンデレラ城ミステリーツアー度：★★★★
アリスのティーパーティ度：★
非スペースマウンテン度：★★★
非スターツアーズ度：★★★
スプラッシュマウンテン度：まだ見たことナイ

●プロミストランド



我が神が導きたもう約束の地とは?

Yamada Junji
山田 純二

巷で人気急上昇のポピュラスに，はやばやとシナリオ集が登場。西部劇編やブロックランド編などAmiga版からの移植５つと，イマジニアのオリジナル，江戸時代編の全６編が収録されている。まだ全面クリアしていない人もこれは見逃せないぞ。



X68000用　　5"2HD版 4,800円(税別)
イマジニア　　☎03(343)8911

５月に発売以降，巷で大好評のポピュラスにさっそく追加シナリオ集が登場。いままでは神と悪魔の対決という設定のみだったから，この朗報にはもろ手を挙げて喜んでしまったわけだ。

この追加シナリオ集には，インディアンと騎兵隊の戦い「西部劇編」，変な宇宙人同士の戦い「シリーランド編」，童心にかえってブロックとたわむれる「ブロックランド編」，ベルサイユのばら（ふっ古い！）を思い出す「フランス革命編」，未来世界での大手コンピュータメーカー同士の争い「ステーショナリーワールド編」，そしてなぜか武士と商人が戦うイマジニアのオリジナル「江戸時代編」と，６つのシナリオが含まれている。で，このバラエティ豊かなそれぞれのシナリオに合わせて，キャラクターデータもちゃんと変更されている。そのうえ，各面の設定条件やコンピュータ側の思考ルーチンにも変更が加えられていて，オリジナルに比べると結構難しくなっている。というか，敵が強くなっているといったほうがいいな。

んでもって追加シナリオだから，プロミストランドを遊ぶには，とーぜんポピュラスのディスクが必要になる。これを知らないとまさに宝の持ち腐れと化してしまうので注意すべし。

このプロミストランド，ルールや操作法，使える奇跡などはオリジナルのまま，特に変更はナシ。ただ，効果音も同じなのはちょっと残念。プレイしてみればわかるけど，各シナリオごとに特徴があるので，それにあった効果音が欲しくなってしまう。どれをとっても個性がつんつんしているとっても楽しいシナリオなので，戦いの音や沼に落ちたときの音がそれぞれ違っていたら，もっとよかったのに……。

この６つのシナリオのなかで，僕が気に入っているのは，江戸時代編での沼地。まるで，肥だめのような雰囲気をかもし出していて，落ちたらとっても臭そう。敵の民が落ちたときに，僕は今まで以上に，エクスタシーを感じてしまった（ん？　危ないって？)。それでは，69面までプレイしたなかで，僕の気に入った（はまってしまった），はたまた印象に残った３つのシナリオを紹介していきましょう。

そちも悪人よのう

ひとつ目は，江戸時代編。このシナリオは，ところどころに桜や松の木があって，なかなか日本情緒しているところが気に入ってしまった。特に面白いのが城の中庭。よ～く見てみると松の木と玉砂利が敷いてあったりなんかして，細かいところまでやってくれるなあ，イマジニアさん，などとすっかり感心してしまった僕。まだ最初の面だからやりたい放題できるのをいいことに，新しいキャラクターの仕草を堪能しつつ，悪行の限りをつくしてしまった。

まず手始めに，必殺肥だめ攻撃！（うわぁ，ディスプレイの向こうから臭ってきそう）もちろん，ただあちこちに沼を仕掛けるわけではなく，周辺に地震を起こして，



江戸時代編。桜も満開できれいだこと

いきなりですが，ポピュラス大会のお知らせ

夏休みにヒマを持て余している諸君，キミのポピュラスの腕を試すときがきたぞ！　なんでこんな企画が持ち上がったかというと，なんでもポピュラスの原作者であるピーター・モリニュー氏がイマジニアのイキなはからいで８月25日に来日するそうな。で，さすがは原作者，対戦ポピュラスにおいては未だ負けたことがないと豪語なさっているらしい。日本のポピュラスフリークともぜひ対戦を，てなわけで，あれよあれよという間にすっかりこの話が決まってしまったのである。

さてさて，この大会には７つのパソコン雑誌チームとイマジニアの計８チームが出場，おのおのの読者代表（イマジニアは違うらしい）をしたがえてこの大会に挑むわけだ。で，トーナメント形式で戦い，その８チームの優勝者がピーター氏と晴れて対戦，まさにポピュラスの王者決定戦というわけ。対戦期日は８月18日と26または，27日。まず18日に８チームの優勝者を決定，26または27日にピーター氏と対戦する予定。

そこで，だ，我がOh！Xでもゼッタイの自信と意欲のある読者代表を求めている。我こそはと思ったら，すぐさま**官製ハガキ**を買いに走り，住所，氏名，電話番号，そんでもってこれがいちばん大切なワケだが，**CONQUESTモードでの最高面数とそのパスワード**を明記のうえ，Oh！X編集部「我こそはポピュラスの王者なり」係まで送ってほしい。応募の締め切りは８月５日（必着)。場合によっては，編集部で腕前を見せていただくのでウソや人から聞いたパスワードは書かないように。それでは，勇気あるポピュラスフリークの応募を待っているぞよ。

相手の民を引きずり出してから，沼を仕掛けるという極悪非道ぶり。そうすると，家から追い出された相手の民が，ボットンボットン，気持ちいいほどよく落ちる。

そうやってしばらく遊んでいると，相手の土地と自分の土地がつながるので，すかさず自分のシンボルであるまねき猫（相手のシンボルは「たぬき」だったりする）を移動して，民を誘導して敵地に突っ込ませる。当然，仕掛けた沼地は，地震と火山で潰しておく。でないと自分の仕掛けた罠に自分の民がはまってしまうという，間抜けなことになってしまうからね。

そのあとは，侍を作って相手の家に放火させてまわってネチネチと相手を攻撃させていったり，洪水を起こしてもう一度いじめ直そうかな，と思ったけど，あまりにも暗いので結局は最終戦争で勝負をつけて終わりにしてしまったのだった。

ぼくらの願いは世界征服だ！

さて，２つ目は子供の頃よく遊んだ覚えのある，ブロックの世界を使ったブロックランド編。マップが見づらいのが難点だが，これといって難しくはなかった。が，しかし53面！　これがとにかく面倒だった。最終戦争を起こせないので，勝つためには相手を個別撃破していくしかなく，しかも騎士が作れない。なぜかというと，圧倒的にこちらが有利になろうとも，相手を全滅させなければならないので，結局はシンボルを移動させ，リーダーをせっつきながら１つひとつ倒していくという，非常に非能率的な戦法を取らなくてはならないのだ！

攻撃しているときでも，相手はどんどんへんぴな場所に分散してしまうので，鬼ごっこよろしく追いかけ回させられる。そのうえ地面を盛り上げることしかできなくて，土地の整備が難しい。人が増えてくると当然のことながら全体の処理が重くなるため，マウスの誤操作がしょっちゅう起こる。せっかく苦労して作り上げた平地が，ちょっとしたミスで水の泡になってしまったことが何度あったか。

この面はホント，これら悪条件のためにストレスが溜まってしまった。１時間も２時間もマウスをクリックしていると，肩もこるし目も疲れてくる。まあ，それだけに勝ったときには，すごくほっとしたけど。

哀愁のプログラマ

そして，３つ目のシナリオは46面のステーショナリーワールド編。僕がプロミストランドで初めて負けてしまったのがこの面。



まるでオモチャの国のようなブロックランド編



こっちはステーショナリーワールド編

日頃付き合いの深いコンピュータ世界での戦いということで，このシナリオは結構はりきって遊ぶぞ！　と思いきや……。

今までと同じようにシンボルを移動させながら，相手の土地を目指して進んでいたらば，しばらくして相手の火山攻撃。１発目のときは，わりと余裕たっぷりに，コンピュータも頑張っているなぁ，と作られた山を削っていた。が，間髪入れずに２発目の火山攻撃を受けたときにゃ，マウスを握る手がピクリ。３発目には思わず，マジかよとつぶやき，４，５発目には目が座って，必死に復旧作業をする僕の姿があった。

すでに，連続の火山攻撃で泣きそうになっている状態に，さらに追い打ちをかけるように「ブォン」と変な音が。思わず背筋がぞくっとして，マップを捜し回ると，いた！　ガチャピンナイト（このシナリオのナイトは，まるでポンキッキのガチャピンの頭に足を２本付けたようなやつで，その愛らしい顔とは裏腹に，領土を荒らし回ってくれる)。しばらくするともう１匹，さらにもう１匹と今度は,連続のナイト攻撃！もちろん，火山攻撃も休むことなく続いていて，結局はたび重なる敵の攻撃に耐えられず，負けてしまった。

あまりの悔しさにすぐさま再度チャレンジしたが，結果は同じく負け。しばらく呆然として，設定画面をながめていたら，“WATER IS FATAL”の１行に気づき，３度目の挑戦にして，ようやく勝つことができた。わかってしまえばなんのことはない。ナイトは海に沈めてしまえばよかったのだ。ここで初めて，プロミストランドが，難しいと実感した。

500面クリアした人はいるのか？

このプロミストランド，それぞれのシナリオは見掛けはおちゃらけたパロディ。が，中身はなかなか手応えあり。それに初めからやり直すのが面倒臭ければ，オリジナルのパスワードが，そのまま使用できるので（サンプル版では），自分が進んだ面から自由に遊ぶことも可能だ。

欠点としては，キャラクターを変えたことによりマップが見づらくなってしまっていること。ステーショナリーワールドは地面の盛り上がりが滑らかにつながっているし，ブロックランドでは角張った地面なので，どこが窪みでどこが盛り上がっているか，慣れてくるまで区別が難しいカナ。

それにしても，オリジナルでさえ500面あるのに，さらに追加シナリオが出てしまって，単純に考えたら1000面。発売からしばらくたっているとはいえ，はたして全面クリアした人はいるのかな。スタッフでは，祝一平氏と西川善司氏の２人が，400面ちょっとのところで争っているようす。ほかには，200面，300面クリアの人がちらほら。しかしまだクリアした人はいないよう。全面クリアしたらどうなるか，気になっているんですけどね。

総評（天国は楽し）

この追加シナリオ集は，それぞれのシナリオに合ったコミカルなキャラクターがわしゃわしゃと動き回り，見ているだけで楽しくなってしまいます。以前，スペースハリアーで，キャラクターデータを書き換えたパロディ版があったのを覚えているでしょうか。あれはただのお笑いの世界でしたが，このプロミストランドはシナリオごとにそれぞれ因縁の対決を再現していてストーリーを感じさせてくれます。

さて難易度ですが，本文中でも述べたと思いますが，“６面から相手はナイトを作れるようになる”と，一言いえばわかると思います。キャラクター自体は可愛いくて愛敬もあるくせに，やることは手厳しい，まさに可愛さ余って憎さ100倍とはこのことです。先へ進むのは結構タイヘン。体力と時間のある方は，ぜひ挑戦してみてください。

総評（５点満点）

キャラクター	5
肥だめ	5
変身	4
シナリオ	4
難易度	5
やっぱり面白い	5

AFTER REVIEW

今月は，天下統一，ダウンタウン熱血物語，あーくしゅの3つに加え，6月号の付録ディスクに収録したYet Another Columnを紹介します。さあ夏休み，思う存分ゲームにひたれるときが来たぞ。この夏やりこんだゲームの感想をどんどん送ってちょ。

天下統一

▶最後の最後まで手が抜けない。最後までライバルといえる勢力が存在する。
岡山県・水口　仁郎(21)

▶私は日本史が好きです。
大阪府・加藤　弓弦(22)

▶現在のX68000のシミュレーションゲームでいちばん楽しめる。
徳島県・中沢　賢一(22)

▶戦国時代のようすをみごとにシミュレートしているから。　広島県・平本　裕司(18)

▶コマンドはかんたんだがよくできている！
千葉県・根市　浩(27)

▶反射神経を必要としないし，自分の住んでいる国から統一にかかれる。
滋賀県・小池　清(42)

▶末長く遊べそうだから。
新潟県・保科　康広(20)

▶画面よし，音楽よし，内容よし。
京都府・可児　典明(17)

▶アルシスの移植と聞いただけで……。



熊本県・中村　巧(19)

▶ほかの機種で有名であったが，それがまたいちだんとパワーアップして登場。
鳥取県・安岡　正美(18)

▶戦いが城単位だから戦略的に自由度が高いのが，思ったより面白い。
北海道・近江　弘和(18)

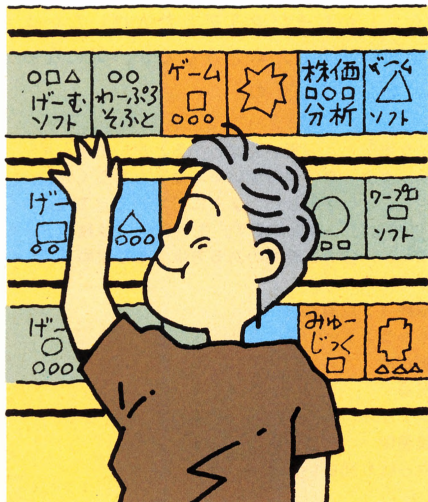
▶戦国シミュレーションファンにはオススメ。
東京都・金子　博政(24)

▶かゆいところに手がとどく
北海道・釜蓋　実(19)

あのアルシスソフトが移植をして，システムソフトが発売した戦国シミュレーションとあって，発売される前から評判だったこのゲーム。フタを開けたらやっぱりこのとおり，の人気でした。フルマウスオペレーションもさることながら，やはりシンプルかつわかりやすい点が，ユーザーの共感を得たのでしょう。統一を目指していく手段も，国対国の争いではなく，1つひとつ城を攻略していくといったやり方なので，ゲームを進めていくうえで，非常にやりやすくできているといえます。また，余計なものを排除したとはいえ，各々のグラフィックもなかなか見応えがあるものでした。しかも，評価版に比べて製品版はかなりスピードアップしているようです。

こういったシミュレーションものは，まず第一にコンセプトがしっかりしているかどうかにかかっています。シンプルでもいい，面白いものを，というその意気込みがひしひしと感じられ，プレイする側としても，うれしい作品でした。

X68000用　5″2HD版2枚組　9,800円(税別)
システムソフト　☎092(752)3902



発売中のソフト

★ギャラガ'88

電波新聞社の今度の新作は，ナムコの「ギャラガ'88」。「ギャラガ」というゲーム自体は1981年に発表され，未だにゲームセンターなどでちょくちょく見かけるが，このギャラガ'88は，1987年に発表されたそのリメイク版だ。自機を2連結・3連結させて，ギャラガ星人を心ゆくまで吹き飛ばしてちょうだい。X68000版には電波オリジナルのボーナスステージが追加される予定というから楽しみ。

X68000用　5″2HD版2枚組　8,200円
電波新聞社　☎03(445)6111

新作情報

★遊撃王II

21世紀の近未来の空に展開する，最新鋭戦闘攻撃機「MI-C.A.D.O.II」型の活躍を描くフライトシミュレータ。ミッションプレイングモードのほかにフライトシミュレートモードが用意され，まず訓練飛行・模擬戦闘でパイロットの腕を磨くことができる。MI-C.A.D.O.IIに慣れたらミッションプレイングモードに挑戦。迎撃，偵察，攻撃，護衛の中から任務を選ぶ。弾数や燃料を考慮し，みごと任務を遂行できれば昇格できる。目指せ，最高階級！　サイバースティックにも対応し，フライトシミュレータファンにはたまらない一作といえそうだ。

X68000版　5″2HD版　予価8,800円
システムソフト　☎092(752)3902

★Thrice

立て続けに新作を発表しているM.N.M. Software。今度はパズルゲームが登場だ。ブロックが上から降ってくるというのはお決まりだが，着地してから回す倒すひっくり返すの大騒ぎ。テトリスでもない，コラムスでもない不思議な感覚のゲームだ。隠れフィーチャー，季節感のあるグラフィック，古代裕三氏のBGM，ビデオ機能に300名までのランキングと盛りだくさんに詰めこんだ，M.N.M.入魂の一作。

X68000用　5″2HD版　価格未定
M.N.M. Software　☎0423(60)3084

★サイバリオン

マニア垂涎のマト，あのタイトーのサイバリオンが家で遊べるようになるぞ。

メカニカルな龍をトラックボール(X68000版ではキーボードなども可）で操り，炎で敵も弾も振

ダウンタウン熱血物語

▶お店へ入っているときのくにおやりきがかわいい。戦い方がいろいろあっていい。

長野県・山崎　芳照(15)

▶画面がどう考えてもX68000のものとは思えないが，やってみるとやみつきになる。

茨城県・関根　信男(17)

▶とんでもないマップさえなければ，最高なのになぁ。　東京都・高見　創(19)

▶ファミコンの移植だからダメかなっと思ったら，これが意外と面白いのよ！

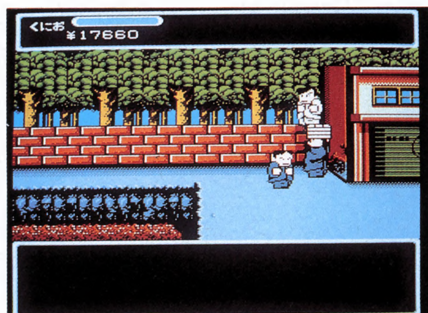
高知県・井上　哲郎(24)

▶他人がどう言おうと私は好きや。

大阪府・渡辺　雅之(29)

たくさんのアイテム，殴る，蹴るなど日頃のうっぷんをはらすにはもってこいだったこのゲーム。やはり，そのあたりの単純さがよかったのかもしれません。マップはやや入り組んでいましたが，それがかえってこのゲームを面白くしたともいえるでしょう。

X68000用　5″2HD版2枚組　8,800円(税別)
シャープ　☎03(260)1161



あーくしゅ

▶ピクトのまじめさに対し，じぇだのすっとぼけた会話がすごくいい。

埼玉県・奥村　光雄(15)

▶じぇだが二重人格者だから。おまけに言うと，マウスカーソルはヤマトとウルトラマンとやじるしもあるぞ。

北海道・谷口　有香(21)

▶短時間で解けるのがいい。

東京都・合屋　琢(21)

このゲームに関してはカワイイ，とか面白いとかいったひと言で表せるような感想が多かったですね。いままでのウルフ・チームとはひと味違って，遊びの部分でできあがっているような感覚が，気負いを感じさせずかえってよかったのかもしれません。それに，なんといってもキャラクターがみんなかわいい。いずれにしても，ウルフ・チームは，こういったパロディものでも，シリアスものでも作れるという実力を見せつけた作品でした。

X68000用　5″2HD版3枚組　6,800円(税別)
ウルフ・チーム　☎03(5273)4795



Yet Another Column

▶Yet Another Columnにハマッています。テトリスの4段消しのときよりも，Yetの連続して消えていくときの気持ちのよさといったら，もう言葉では表せません。得点は3万点ちょっとなので，努力して4万点突破を目指すぞ！　静岡県・富永　恵隆(19)

▶な，なんなんだYetのあのスピードは（速くなったときのことだよ)。バカヤロウ，キーの反応が追いつかないくらい速く動かすんじゃねー！　愛媛県・柳井　敏彦(31)

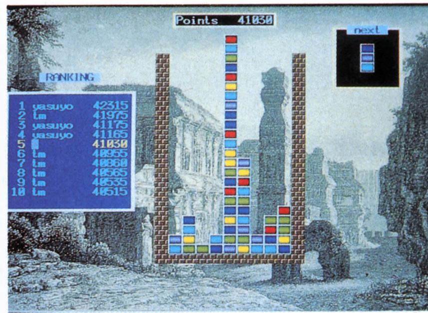
▶テトリスより熱中してしまった。ヘタに込み入ったゲームよりもシンプルで，なおかつ面白いのはこいつくらいだろーな。感謝であります。　沖縄県・大城　久(18)

▶思いがけない連鎖反応が好き。

福岡県・村上　淳一(18)

まあ100号記念ということで，こんなのも今回は入れてみました。付録でつけたとはいえ，好評を得ているのは編集部としてもうれしい限りです。みんな，遊んでくれてますか？

Oh!X1990年6月号付録ディスクに収録



り払ってこれまたメカニカルなボスキャラと対決する。実戦モードでは，プレイごとに独自のシナリオと独自のマップが作られ，ストーリー展開に従ってパワーアップしたり無敵化したりする。おまけにボスキャラのなかには，「ダライアス」のボツキャラも入っているとか。操作感覚に慣れるための練習モードもあるぞ。

移植はSPS，トラックボール対応とくればいやがうえにも期待は高まる。今からトラックボールさばきを鍛えておこう。

X68000用　5″2HD版　価格未定
シャープ　☎03(260)1161

★ラグーン

言わずとしれた「ジェノサイド」のズームが放つファンタジーRPG「ラグーン」がいよいよ発売になるぞ。300年前。7人の魔導士が邪神を呼び出してしまったことがすべての発端となった。3人の命を犠牲にして邪神は封印されたものの，この一件は魔導士の間に決定的な影響をもたらした。邪神の力に魅入られ，その力を手に入れるべく「闇の皇子」を捜す魔導士ゼラー。そしてその闇から世界を守ろうと「ムーンブレードの勇者」を捜す魔導士マティアス。そして彼は少年ナセルとの決定的な邂逅を果たす。彼こそがムーンブレードの勇者なのだ……。

子供が泣きだすほどのデカいキャラと，ゲーマーが腰を抜かす激しいアクションに酔いしれてちょうだい。

X68000用　5″2HD版　8,800円
ズーム　☎011(613)0191

★幻獣鬼

古より，6つの魔界との接点「結界」に囲まれた王国ジタンの人々は魔物と戦う宿命にあった。しかし，ある日無能な魔導士が結界を破り，魔物が王国に攻めこんできてしまった。戦士レオン，魔導士リィノ，忍者ルイカの3人は，結界を封じる6つのロシュファの魂を奪い返すために旅立つ。MSX専用に開発された「アンデッドライン」が，X68000用にパワーアップしてリリースされる。プレイヤーは3人のキャラを自由に選び，好きなステージから攻略してゆく，キャラによって面のアイテムなどが微妙に変わるなど，数々の趣向を凝らしたT＆Eの自信作だ。

X68000用　5″2HD版　価格未定
T＆E SOFT　☎052(773)7770

★イメージファイト

つぎつぎとビデオゲームの移植が続いているなか，ついにシューティングゲームの真骨頂，イメージファイトが登場。20XX年，東西陣営の軍事競争のなか，突然西側のムーンベースが大爆発を起こした。西側未確認の戦闘機によるものと判断した西側は，最新戦闘機OF-1を急きょ用意した。訓練飛行は完全ではないものの，コンピュータシミュレーション試験に合格した者は即，宇宙に飛び立っていった。

最初の5ステージがそのシミュレーション面になっており，平均90%の撃墜率をマークしたものだけが実戦へ進むことができる。落第者は補習ステージ行きだ。ポッドシュートやスピードチェンジ，特殊攻撃パーツを使いこなし，目指すはムーンベース内のマザーコンピュータだ！

X68000用　5″2HD版　価格未定
アイレム販売　☎06(535)4880

★バルーサの復讐

反響を呼んだ「トリトーン・ファイナル」の続編だ。剣と魔法を駆使する8方向多重スクロールのアクションだ。大魔王アレスターに侵略され，国を捨ててウオークの国にやってきたひとりの少女。彼女はウオークにくる途中，突然現れた悪魔により，船は難破し，彼女の兄は呪いをかけられ連れ去られてしまったという。勇者ステイルは，大魔王アレスターの持つムーグ石で少女の兄の呪いを解くため旅立った。

X68000版　5″2HD版　価格未定
ザイン・ソフト　☎0794(31)7453

ADVANCED 2D GRAPHICS

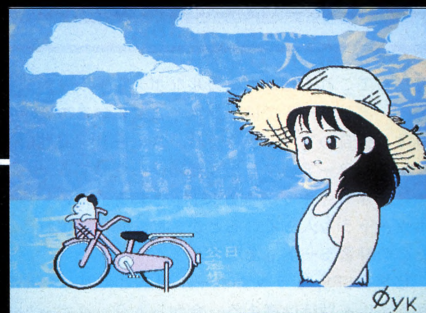
X68000のX-BASICに初めて触ったときのことを思い出す。アナログRGBをサポートしたマシンのグラフィックは……と期待しつつLINEを引いて，表示されるギザギザした線にちょっぴり失望したものだった。

これまでのグラフィック特集ではどちらかといえば３Ｄ処理を主体にしていたように思う。これもX68000発売から比較的早期にZ'sSTAFF PRO-68Kが発売されたことが大きい。このツールはそれまでのパソコングラフィックの枠を超えた処理を実現した。そして名実ともにX68000の標準的グラフィックツールとなっている。

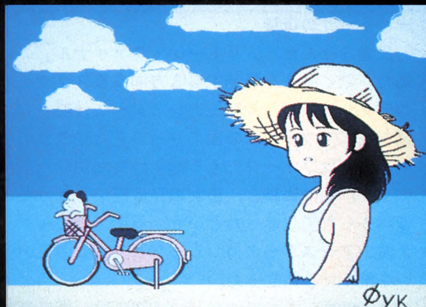
しかし，はや３年。内容はともかく，もはや新しいコンセプトのツールとはいえない。その他のツールもZ'sSTAFFに追いついていない。もっと違ったコンセプトに基づくツールができてもいいのではないか？



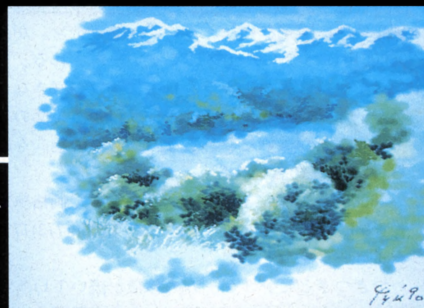
これをスクリーントーンとすると……

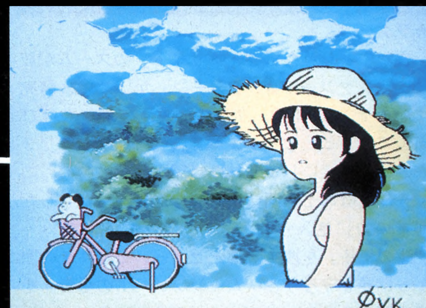


スクリーントーンつきペイント

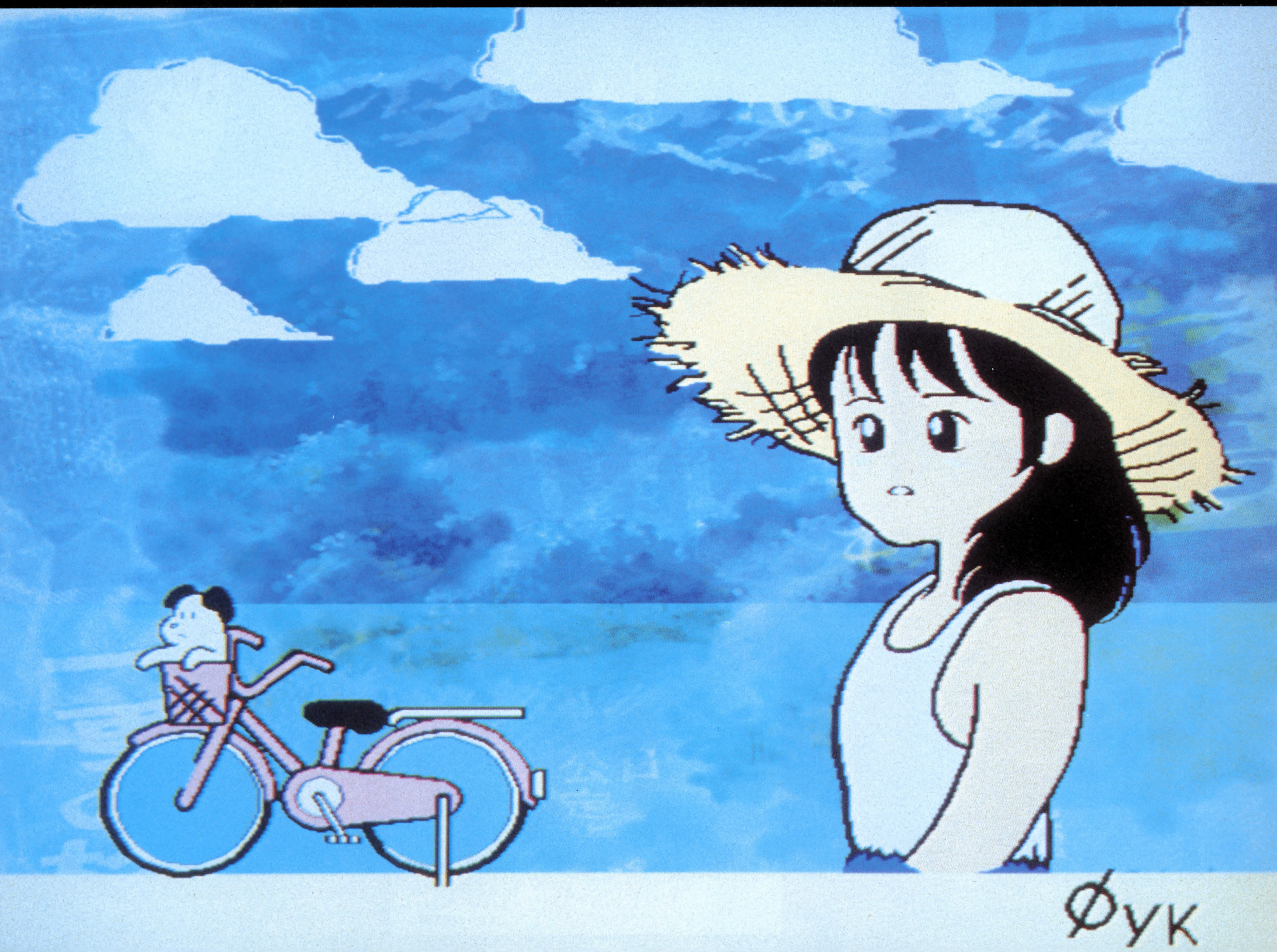


元絵


タイルとして登録し……



タイリングペイント



アンチエリアシング対応スクリーントーン&タイリングつきペイントルーチンの応用例。タイリングペイントとはいってみればグラフィックパターンの連続張り付けだ。デジタルRGBでは多色表示のために使われていたが，アナログRGBではあまり使われない。メモリに余裕があれば張り付けるタイルの大きさに制限をつける必要はない。これはヘッダを書き換え，最大512×512ドットの画像をタイル登録できるようにした関数での実行例だ。空の部分にスクリーントーン（全画面分の新聞紙）とタイル（背景）をペイントした。ビデオなどのクロマキー合成に似ているが，アンチエリアシング対応なので，マスキング不要で本当に塗りたい部分の隅々までペイントできる。スクリーントーンとは合成の比率を決定するもので，パターンさえ用意すればぼんやりとオーバーラップする画像や任意範囲の階調つきマスキングにも使える。

CONTENTS

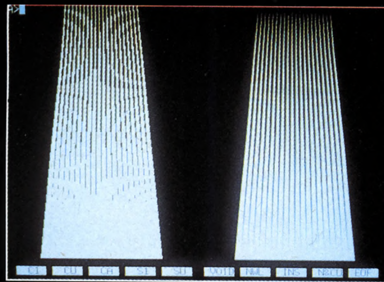
X68000用グラフィックツール紹介
あなたにあったグラフィックツール ……荻窪 圭 44

ギザギザのないグラフィック関数
アンチエリアシングとは? ……丹 明彦 50

X-BASICによる画像処理
後処理によるジャギーの除去 ……中野修一 68

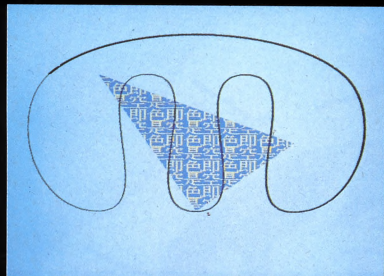
色数の補間と量子化
グラフィックを変換する ……鈴木康弘 72

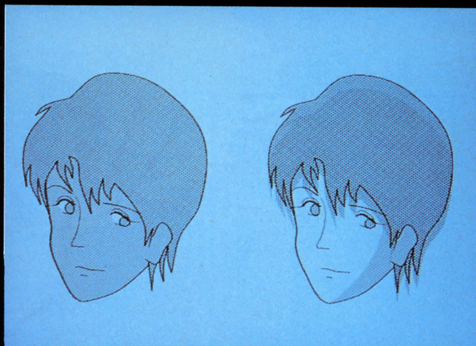
4096色→8色変換
Zの画像をX1で ……亀田雅彦 77

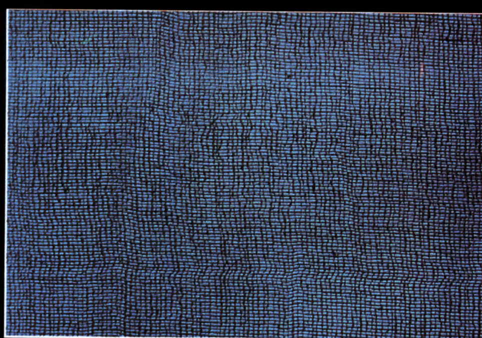

滑らかなラインを見よ

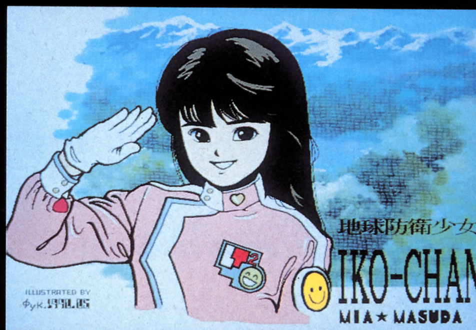

従来の関数による画像


アンチエリアシングされた画像


半透明のスキャンコンバージョン


スクリーントーンを使う


ガーゼを取り込み反転する



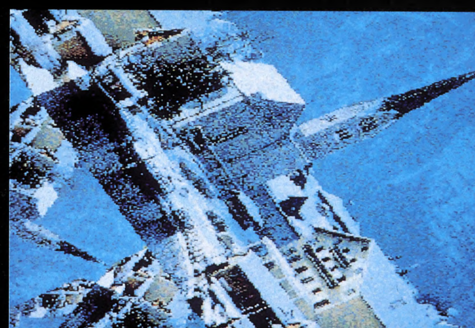
キャンバス地のような表現となる

丹明彦氏によるX-BASICで使えるアンチエリアシング対応のグラフィック関数の使用例。滑らかなライン（ライン幅調整可能），ベジェ曲線による滑らかな曲線，そしてタイリングとスクリーントーンに対応したスキャンコンバージョン（閉曲線領域の塗りつぶし）とペイントルーチン。すべてがアンチエリアシングによる多階調の境界線に対応している。
これらの新しい関数群は単にいままでのBASICにあった関数の発展版としても使えるが，柔軟な思考で使い方を変えれば，さらに新しい可能性が見えてくるはずだ。すでに前ページで行った画面合成。機能が柔軟ならペイントでこういった処理までできてしまう。スキャナを使ってガーゼを取り込んだものをreverse(　)で反転し，tone_get(　)でスクリーントーンとして登録。左の写真のキャンバス地のような背景はこのようにして作られた。
2Dグラフィックもまだまだ面白い可能性を残している。

XROT0によるグラフィックの回転

特集の記事ではないが，読者投稿によるグラフィック回転プログラムの実行例。短いプログラムでしかもかなり高速。サンプルプログラムはキー操作により拡大縮小自由自在でぐるぐる回転する。
画面下の領域に画面の内容が再帰的に反映されているのも面白い。デモやゲームの特殊効果はもちろん，グラフィックツールの一部として使っても面白い機能だ。


これが元の画像


回転後。下に再帰している部分が見える

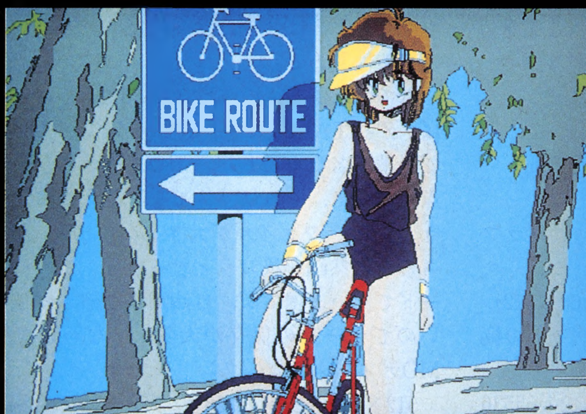
PC-9801のアナログ16色モード用に描かれた画像をX68000の65536色モードに変換した例。もちろんアナログパレットにも対応し，タイリングパターン部分を単色のベタ塗りに変換する。
画面サイズを補正して512×512ドットの全画面対応に変換した場合は輪郭線の一部が切られてしまっている。横640ドットにしたものはプロポーションが原画と異なっているが，ドット構成は正確に変換していることがわかる。CRTCをいじれば横768ドットのまま正方形に近いピクセルで65536色を出すことも不可能ではない(自由研究)。
タイリングが単色に置き換えられるため,65536色化されたもののほうが元ファイルより圧縮がきく場合も多い(PIC.R)。なお，原画は森林林檎氏によるもの。



元の画像(640×400)



拡大図　タイリングが見える



512ドットで変換

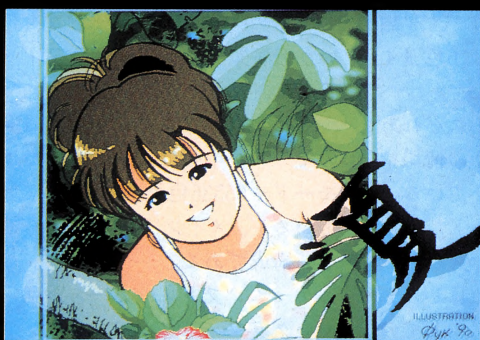


640ドットで変換

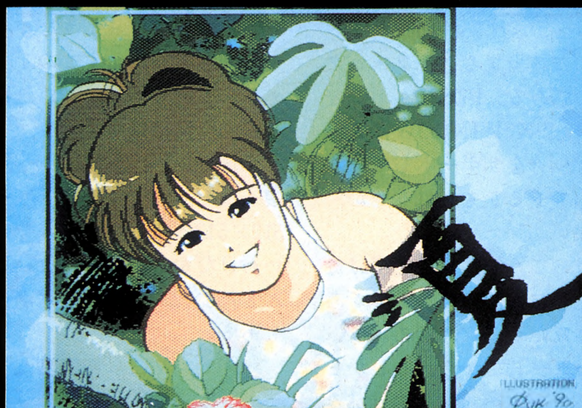


タイリングが消えた

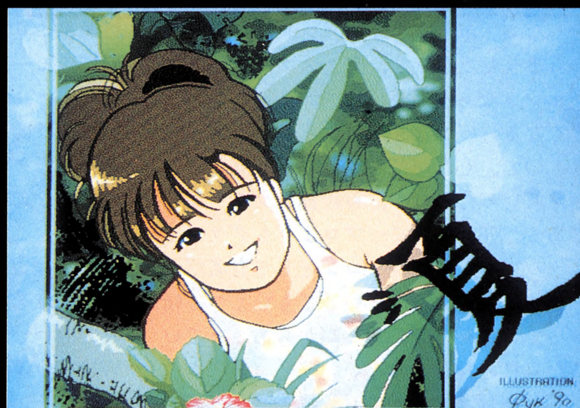
65536色の画像をできるだけ原画に忠実に256色モードに変換した例。オーダードディザ法を使ったものと桒野雅彦氏のアルゴリズムを応用して多色化したもの。
よーく見ないとわからないが比較的規則的なパターンになるディザ法と，かなりランダムなパターンになる桒野式のアルゴリズムの違いがパターンになって表れている。階調表現は桒野式のほうが自然に思えるが，もともと白黒２色用なためか，隣接するドットの明暗差が激しく出るのがやや気になる。多色用にアルゴリズムを改善できるのではと思う。誰か挑戦してみてほしい。



これが65536色



オーダードディザ法



これは桒野式

X68000用グラフィックツール紹介

あなたにあったグラフィックツール

これまでにX68000用として発売されているグラフィックツールを集めてみました。それぞれの個性や使い勝手について独断と偏見を交えて試用レポートをまとめました。皆さんのツール選びの参考になるでしょうか？　それではサンプルは電脳絵師の福原徹でお送りします。

Ogikubo Kei 荻窪 圭

よくこーゆーことをいうやつがいる。
「それで，なにが面白いの？」
ノートに落書きしてるのを見ても絶対そんなことを聞いてきたりはしない。
「で，さあ，それって，役に立つの？」
役に立たなきゃいけないときたか。
「プリンタで打ち出したりできるの？」
できねえよ（画面と同じようにはね）。
やっと，そいつのいうことがわかった。紙に描いてあったり，ビデオで見られたりしない絵は価値がないというのだ。

グラフィックを描いて遊ぶなんていうのは，コンピュータはなにか役に立つもの，と信じている善良な市民にとって信じられないことらしい。そう考えてみると，グラフィックツールで遊ぶなんてのは，かなり贅沢な道楽のようだな。道楽万歳。

＊　＊　＊

目の前に山があるからといって別に登りたいとは思わないが，目の前にあるのが紙とペンだったりするとなにか描きたくなり，楽器だったりするとなにか音を出したくなる。誰も文句はいうまい。目の前にあるのがポピュラスだったりすると締め切りも忘れて沼を作りたくなるというバリエーションもあるぞ。

それでもって，目の前のパソコンにFM音源やAD PCMが乗っていれば鳴らしたくなるし，65536色出るとわかれば色を出したくなる。ポップアップハンドルがあれば持ち歩きたくなるし，ディスクがオートイジェクトならゲットイン/ゲットアウトしたくなる，ってなもんだ。それが人情というもので，それが楽しいわけである。

そういったわけで，お絵描きソフト集合である。X68000はワープロよりもグラフィックツールが多いパソコンとして有名だが，グラフィックツールといってもいくつも転がっているわけで，片っ端からあさっていったら私の身がもたない。

で，今月は2次元のお絵描きソフトである。2次元のお絵描きというのはつまり，CRTに投影されているグラフィックVRAMをぺたぺたとデータで埋めていくことを目的とした作業のことだ。これがグラフィック画面で遊ぶ基本。

グラフィックモードへの対応

X68000の場合，ご存じのとおり，グラフィックモードをたくさん持っている。

まず1024×1024（表示画面は768×512）の16色。ドットが小さくて，1ドットの縦横比がほぼ1：1である。

続いて，一番メジャーな，512×512の65536色。1ドットを16ビットで表現しようという贅沢さで，512Kバイトのグラフィック RAMがたったひとつの画面に収まってしまうという恐ろしいモードである。

さらに，意外とおいしい512×512の256色。1ドットを8ビットで表すわけで，2画面分持てる。さらに，512×512の16色（4画面だ）。

その下に，256×256モードがそれぞれあって，このモードは1画面当たりの情報量が少ないため，高速な処理に向いている。シューティングゲームに多いモードだったりする。

とまあ，こんなにあるわけで（ほかにもいろいろ隠れてたりするけれど），すべてのモードに対応しているグラフィックツールなんてない，のだ。

順番に見ていくと，まず756×512ドットの16色モード！　に該当するグラフィックツールは，なし，である。PDSにもあるという話は聞かない。SX-WINDOWはこのモードのグラフィックをサポートしているので，そのうち出てくるかもしれないが，いまのところ，ない。

これはこれでけっこう綺麗な絵を描けたり，文字を埋め込むには向いているのであってもいいと思うんだが，ないなあ。SX-WINDOWがこのモードだから，もしかしたら，そのうち，マックペイントの玩具みたいなのが出てくるかもしれない。また，PC-9801のグラフィックのちょっと大きいやつだと思えば，また違ったものが出てくる可能性もある。

さて，512×512ドットの65536色，といえば，Z'sSTAFFと，G68Kである。X68000で一番有名なモードだ。このモードにも欠点があって，それは「メモリをたくさん食う」とか，「ファイルが大きくなる」だ。もちろん自然画を扱おうと思ったら，このモードでないと困るが，自然画は圧縮しづらいのでデータの保存が大変。てなわけでMOディスク万歳。

512×512ドットの256色。実のところ，手描きであれば，このモードで十分な気がする。そこに気がついたのがサン・ミュージカル・サービスであって，マジックパレットという軽快な異色グラフィックツールを出してきた。開発がサン・ミュージカル・サービス，発売がミュージカル・プランという音楽業界コンビのグラフィックツールである。

それから，ウルフ・チームのPRISMもこのモードが中心だ。一応こいつは256×256モードや65536色モードなどもサポートしているが，メインは256色。ゲーム屋さんらしい構成である。

ゲームソフトメーカーというのは，ついついグラフィックツールを出したくなるようで（そりゃあ，社内で使うために作ったものがあるはずだし），ザイン・ソフトからも予定されているようだが，間に合わなかったのでとりあえずこの4本だ。テラッツォなんてのもあるが，あれはスプライト系なので今回ははずす。

X68000の主なグラフィックツールはこの4つだ。256×256ドットモードのときは，512×512モードの左上4分の1を使えばいいわけだから，問題はない。なかにはちゃんと256×256モードをサポートするツールもある。

画像フォーマット

続いて，とにもかくにもグラフィックツ

ールを使ううえで問題となるのが画像データのフォーマットであった。いくらたくさんツールがあっても，それぞれみんな勝手気ままなフォーマットでセーブされたら，たまったもんじゃない！　ってことは，有史以前からいわれていた。クスコーの壁画にも書いてあったほどだ。

X68000の場合，非常に幸運なことに，3つの標準的なフォーマットがある。そのうちの2つはたいていのグラフィックツールがサポートするというラッキーな結果だ。

第1がGL3（65536色モード時）フォーマットである。ベタフォーマットともいう。X-BASICのIMG_LOAD，IMG_SAVE関数で読み書きできるフォーマットであって，X68000ユーザーなら誰でもこれでセーブされたグラフィックを読むことができる。

ちなみに，256色モードではGM3，16色モードではGS3，256×256ドットモードでは3番目の数字が0になる。

この方式の面白いところは，セーブされた画像のモードをファイルの拡張子で区別していること。ファイルには画像データしか入っていないのだ。つまり，どのモードでセーブしたかがデータを見ただけではわからないのだ。私はこういうのはアナーキーで好きだが，無秩序で嫌いだという人もいるかもしれない。この方式をサポートしていないのは，上の4つのうち，Z'sSTAFFとPRISM。G68Kにいたっては，GL3フォーマットを標準フォーマットに採用している。

ちなみに，この形式はもちろんデータ圧縮をしないため，512×512の65536色だと1枚セーブするたびに512Kバイトの磁性面を消費する。ディスク1枚に絵が2枚しか入らないわけだ。

第2が，ZIMファイルである。これは，とにかく権威のZ'sSTAFFである。X68000用で初めてのグラフィックツールで，あまりにメジャーなため，あとから出したソフトはたいていこのファイルを読む機能なり自分のソフトのフォーマットに変換するツールなりをつけることとなった。

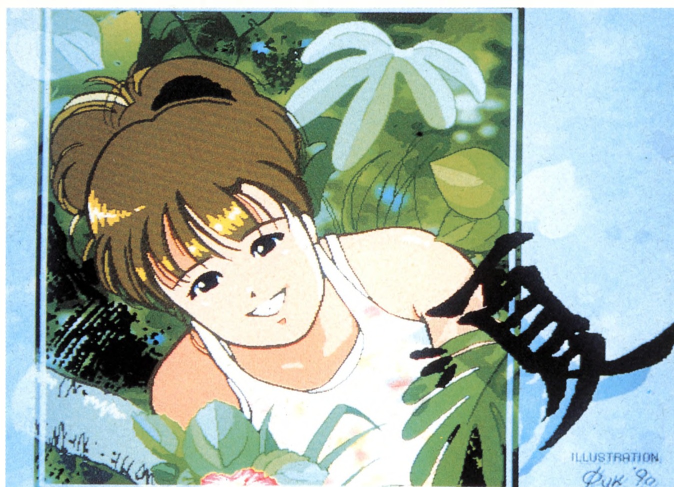
圧縮形式と非圧縮形式があり，たいてい非圧縮形式をさす。ZIMファイルはX68000に向いているかというと，そうではないという意見が大半を占めていて，評判はあまりよくない。

Z'sSTAFF（当たり前だ）のほか，G68K，PRISMがサポートしている。

3番目がPIC形式。PIC.RというPDS（正しくはフリーウェア）の画像データ圧縮・展開ツールの形式だ。圧縮効率が高いのが好まれるところ。でも，PDSなもんで，市販のソフトで対応しているものはなかったりする。自然画を使うのでないならば，とても有効だ。

しかし，どのグラフィックツールも，画面にロードした絵を消さないで起動する方法があるので，ファイルコンバートよりも，こいつを使ったほうが楽だったりする。

では，ひとつずつ簡単にレビューしていこう。



Z'sSTAFFによる作画例

Z'sSTAFF PRO-68K

とにかく，あまりにも有名。PC-9801用のZ'sSTAFF KID-98やらX1turboZについてきたZ'sSTAFF Zからの伝統芸は衰えるきざしなし。伝統の重みはX68000にまで及び，PC-9801なんかと互換性のあるファイルフォーマットを持ち込む（ZIMファイルと呼ばれる）という荒技に出たが，それが唯一の欠点らしい欠点である。

これについては，恐怖の常駐ソフトPIC FILERなるPDS（正しくはフリーウェア）が電脳倶楽部に掲載され，ひとつのマニアックな解決を見せている。これはPIC形式ファイルのロード/セーブをZ'sSTAFF上から行うものだ。

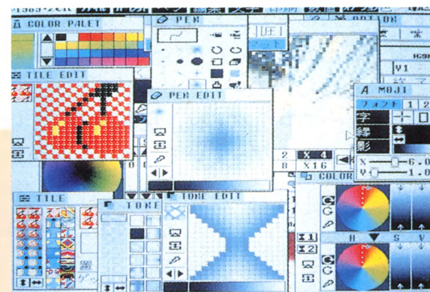
次のバージョンではPICとはいわないが，GL3形式のロード/セーブくらいはサポートがほしい。

操作の基本は，プルダウン風のメニュー。メニューバー上のメニュー，ファイル，パレット，ペン，編集，文字，印刷，数値，オプションの8つから必要なものをクリックすると，ぼよんとウィンドウが開く。その気になれば，描いたグラフィックが全部隠れるほどウィンドウが開きまくる。

グラデーション，トーン，タイル，にじむ色，自由なペン先，スプライン曲線などお絵描きの機能はやたら豊富。

特にそのグラデーションパワーはライン，ペイント，ボックスフィルや閉曲線ペイントなどいつでもどこでも使え，誰でも描ける富士山とか誰でも描ける円柱などの技を作り上げた。

ペン先やブラシだけでなく，ポップで派手なタイルやトーンなどほとんどのものが編集可能で，特においしいのが濃淡の調節である。ペイントやカラーチェンジも，指定範囲内の色に対して行えるので，各種効



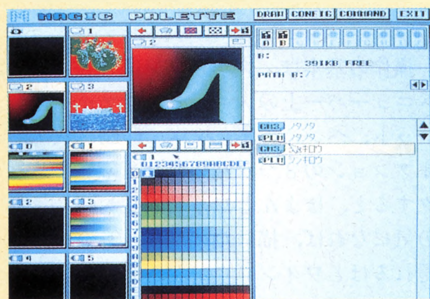
メニューは画面一杯開いてまだ余るくらいたくさん開ける。下がPICFILERを使ったところ。本体のみでも自由変形に色変換と機能は尽きない。強いて欠点をいえば，マスクのみのセーブができない，2枚の絵を重ねる機能などがないというところか

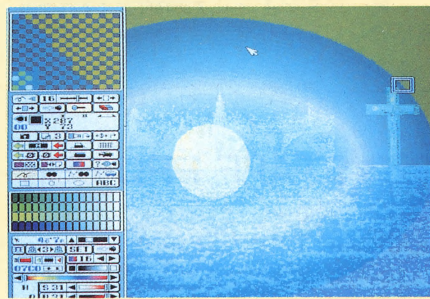






鉛筆画っぽいイメージを目指してみた。セピア調でパレットを統一し，極細ペンを使ってマウスでごりごり……。256色512×512モード固定ながら，なかなか多彩な機能があって使い慣れれば相当器用な絵も描けるのではないかと思わせる。ほとんどの機能がメインウィンドウに収まっているのでわずらわしさがない。消しゴムもいい。





果が狙える。

編集機能も任意矩形の回転・変形・拡大・縮小，任意曲線内のムーブ・コピー。気になるのは，ムーブしたあとに残る白い跡。背景色が白になっているためだ。

外部入力についてもスキャナからカラーイメージユニットまで対応している。バージョン2からはJIS第1水準のみだが，明朝体とゴシック体のアウトラインフォントもサポートされ，X68000ではどのソフトよりもきれいな漢字が書ける。グラフィックに淡色のグラデーションアウトラインフォント文字を入れると，実に気持ちがいい。

おっと，忘れていたが，一部では致命的ともいわれた「プロテクトモジュール」によるコピープロテクトは，現在発売しているものにはなくなっている。買ったらついていなかったので驚いた。よいことだ。プロテクトモジュールつきのバージョン2.0を買ってしまった人は残念でした，と。

欠点といえば，プログラムがでかいため，メインメモリが2Mバイトないとアンドゥ機能が使えないことと，アウトラインフォントを使おうと思ったら，ハードディスクがないと大変だということくらいだろう。

512Kバイトの広大なメモリをアンドゥするのは大変だとは思うが，2MバイトでもRAMディスクをとったり，変なものを常駐させたりしていると駄目である。フリーエリアが1.5Mバイトくらいあれば大丈夫だ。

それから，右ボタンで途中の作業をキャンセルするのだが，「ひとつ前の状態に戻るのではなく，その機能自体がキャンセルされてしまう」のはいただけない。

まあ，どっちにしろ，機能と表現力ではまだ他の追随を許さない。Z'sSTAFFの天下はまだ続きそうだ。

マジックパレット

256色モードに目をつけただけでなく，ペインティングソフトとしてのインタフェイス構造も新しい。ファイル入出力用メニュー画面。ワープロやエディタみたいなカット&ペースト。アンドゥ用メモリ。メインメモリを2Mバイト積んでいれば，チャイルドプロセスでコマンドシェルを起動できたり，描画画面を3面持てたり。

円のグラデーション（外周から中心へのグラデーション）が派手なおかげで，ほかにもあるユニークな機能は見落とされがちだが，アンドゥ用メモリから任意の形で前のデータを切り出せるとか，カット&コピーバッファも編集できるとか，パレットコード&H00を透明色に固定し，背景の基本を透明色にしていること（画面の重ね合わせに便利）などなど。

特に背景が透明色だというのは嬉しい。どこでどう間違ったのか，絵は白い画面に描くもの，といった重力に魂を引かれたソフトが多いからだ。

まず，ファイル入出力モードで立ち上がる。3画面＋コピーバッファ，そして6つのパレットに入れたいファイルがあったら読み込むのである。終了時もこの画面に出て，セーブするなりする。デザインはとてもよい。

そこからコマンドシェルを起動することもできる。drawを選ぶと縦長ででかくてデザインを優先したようなポップなウィンドウが現れる。アイコンがたくさん並んでいて，カラフル。ウィンドウは3つに分かれており，上1/3が描いたりコピーしたりするもの。まん中がパレット。その下がパレット関係の処理。

たとえば，グラデーションバーの両端に



絵の一部をペンとして使う

お絵描きツールのユーザーインタフェイス

いつか祝センセが書いてましたが，ユーザーインタフェイスというものは，たとえ操作しやすくなったとしても，それが古いタイプのものよりも格段にメリットがない場合，人はわざわざ新しいほうに移らないものだそうです。

X68000の場合，最初にZ'sSTAFF PRO-68Kという強力なツールが発表されていましたから，後発のソフトは信者獲得には辛いものがあったろうと思われます。僕自身がZ'sSTAFFの虜となっているので，今回の寸評もそこからの視点を中心に書いてしまっているのではないかと少々不安もあったりします。

が，正直なところ，僕はZ'sSTAFFのようなウィンドシステムは好きではないのです。「下が見えなくて邪魔」なのが主な理由です（これは開発者も感じたらしく，Ver.2ではウィンドウが若干小さく変更されていました）。

グラフィックツールにウィンドウシステムはあわない気がします。かといって，ウィンドウ以外に機能を使いやすく配置する方法っていうのが，まだわかっていないんですよね。描画画面を小さくして周りに配置してしまうってのも手でしょうけど，画面を有効に使えなくて悲しいし……。いい方法はないでしょうか。　(T.F.)

色をセットして，そのあいだの色の変化パターンをいくつにするか決める。それでもって，パレット上のそのグラデーションをセットしたいところへ置くと，ずらっとグラデーションした色がパレットに置かれるのである。マジックパレットでいうグラデーションは，あるパレット番号からあるパレット番号への色の並びにすぎないので（中身の色はなんでもいい），赤黒青緑といった4段階グラデーションもできるし，虹も描ける。

処理の基本は前にも書いたが，カット&ペーストである。任意領域をカットしてバッファへ移し，それを任意の位置へペーストする。バッファを編集したりできるし，透明色を背景にしておくと，重ね合わせが簡単にできる。

その代わり，縮小・回転・変形処理が任意の領域に対してできない。回転や縮小をしたいときは，対象のものだけをほかの画面へ持っていき，画面全体の256×256の画面に対して行ってから，戻すといった作業が必要で，複雑な絵を描こうと思ったら，まめにパーツをセーブするのがいいだろう。あと，トーンの処理も面倒だ。

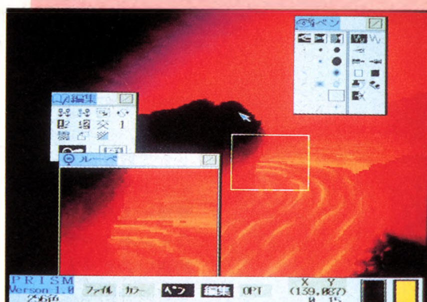
アンドゥ処理はユニーク。アンドゥはアンドゥ用画面メモリから戻されるのだが，そのメモリへのデータ格納は手動なのだ。で，面白いことに，消しゴムを使って画面を消すと，その下にはアンドゥ用メモリの画像が現れるのであった。アンドゥというより，いろんな画面効果に使えそうだ。

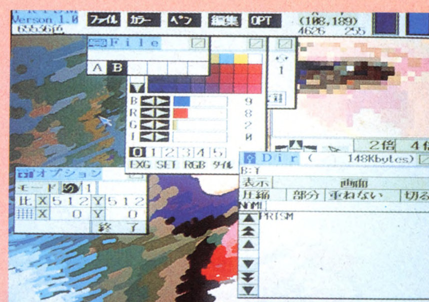
無理をいえば，任意のパレットを使ったカラーイメージユニットからの取り込みか，もっと上手な65536→256色変換がほしい。メニューやコピー時の領域が画面内に制限されているので画面が狭く，ちょっと不便なのも惜しいところだ。

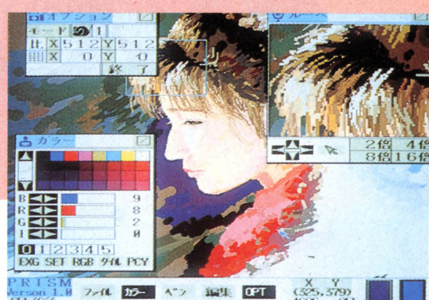
なんだかんだいっても，Z'sSTAFFの影響を免れないソフトが多いなか，こいつだけは違う。非常にポップで軽く遊ぶには最適だ。

16色モードでは画面上の絵をスプライトデータに落とすことも可能だし，起動時にSキーを押しながら立ち上げると直前に走っていたゲームなどのスプライトデータとスプライトパレットを読み込んでくれるのでスプライトエディタとしても使える。

おまけで，マジックパレットのデータをBASICで使うための関数やBASICプログラムのサンプルがついてきて，とても便利である。ついでに，Cのライブラリもあればコンパイルできてよかったのに。オートデモもある。







太めのペンでベタベタと描いてみた。油絵調に見えるかな？（河○純子ちゃんがモデル）マウスボタンの左右に色を設定でき細かい修正に便利。マウスの反応速度を調整できるのもいい。特殊効果に弱いのとスキャナ・プリンタに対応していないのが辛い。

PRISM

こいつはZ'sSTAFFの影響を逃れられなかった。最初から大樹の陰にいたのかもしれない。

ウリは，2色から65536色まで，256ドットから512ドットまで対応した多彩なモードと，アニメーション機能。さすがウルフ・チームである（そーいえば，昔侍ジャイアンツにウルフチーフって選手がいたなあ）。

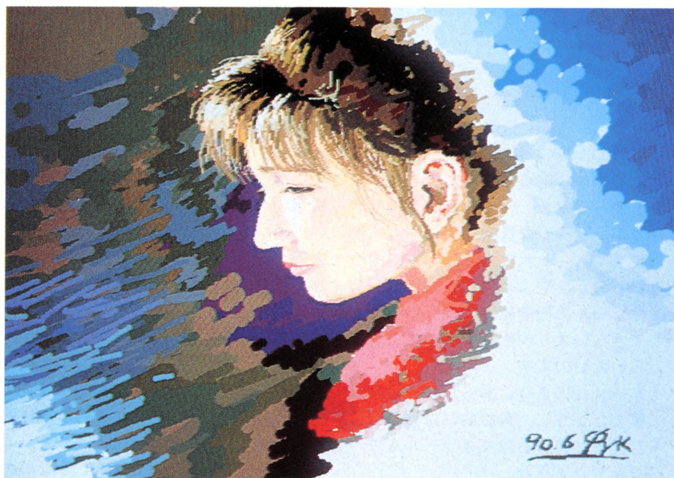
しかし，なんといっても，円が描けないとかグラデーションペイントができないとか文字入力がないとかペン先やブラシの編集もできないとかカラーイメージユニットもイメージスキャナも使えないといった事情にはなにか深いわけでも……と考えてしまう。

その他の操作性は遅いZ'sSTAFFという感じだ。ウィンドウデザインも似ている。ウリはやはりアニメーション機能か。画面上の任意の矩形をたくさん切り出して，連続して見せてアニメーションしてしまおうという機能だ。まずマウスで1コマの大きさを決め，15コマまで任意の位置を切りとって並べる。1/60秒単位で1コマの時間を指定できるから，サブリミナル効果測定テストなんかもできて面白いぞ。どのタイミングが一番いいかテストして，学園祭では売り上げ倍増だ！（そんなにうまくはいくもんか）

画面を2画面まで持てるので片方を背景に使うとかすれば，なかなか，このソフトの意図も見えてくるかもしれない。

ゲームでは特殊な画面モードを使ったりするためか，ふつうのグラフィックツールではサポートしないような512×256ドットモードなどや16色モードなどにも使えるが，その半面，どのモードでもできるような機能しかついていないのが残念だ。とりあえず，どのモードでも絵は描けることを特徴としている。

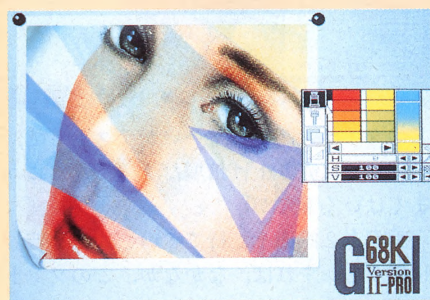
256色モードでアニメーションして遊びたい人は，マジックパレットとPRISMの2つを買って，マジックパレットで描いてPRISMで動かす，ってのもいいかもしれない。定価ベースでは，この2つを買ってもまだZ'sSTAFF PRO-68Kより安いのだ。

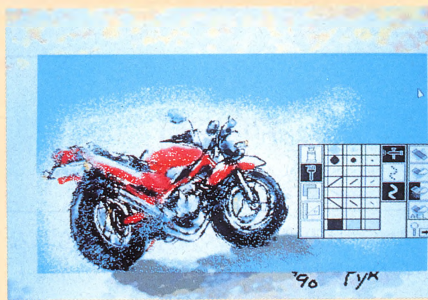


PRISMの使用例。油絵調を狙ってみた



手元にあった鉛筆の落書きを，ハンディスキャナで取り込んでエディットした。水彩画を意識してみたが……。ウィンドウ操作が比較的速いのと，カラーチェンジや閉曲線コピーなど編集機能が多く揃っているのがよい。ただし入出力ファイルの形式がGL3なのが少々不満。ルーペは画面一杯に拡大し，そのままエディットできる。





G68KII Version2.0-PRO

バージョン1では，日本初のBGMつきグラフィックツールという快挙を成し遂げてくれたG68Kであるが，バージョン2ではおとなしい作り（というかまともな作り）になっている。

立ち上げて驚くのが，真っ白な画面にポツンと十字カーソルがあるだけのまぶしい画面。メニューバーからメニュー選択するプルダウン式ではなく，その都度右ボタンでメニューを開いていくポップアップ式なのだ。

昔，（PC-9801の話だけど）Z'sSTAFFと並んで有名だったグラフィックツールにシステムソフトのアートマスターというのがあった。このアートマスターもポップアップメニューで，アイコンやらメニュー構造などが非常に似ている。要は慣れの問題で，開いたウィンドウがうっとうしいという人もいれば，いちいち右ボタンでウィンドウを開くのがうっとうしーという人もいる。

機能的にはグラデーションペイントがないくらいで，普通。

パレットにタイル模様もセットできたりとか，マスク機能が使いやすいといった長所もある。使い勝手の差は，ポップアップメニューが馴染むか否かだろう。

独自のファイル構造や圧縮方式を持っておらず，データはすべてGL3形式というのが素直といえば素直でよい。

Z'sSTAFFをよほど意識しているらしく，Z'sSTAFFの非圧縮ZIM形式とGL3形式の相互ファイル変換が可能となっている。

機能的にはZ'sSTAFFと比べるのがかわいそうだが，価格が半分以下であること，Z'sSTAFFより少ないメモリで動くといった面もあり，一概にはいえない。

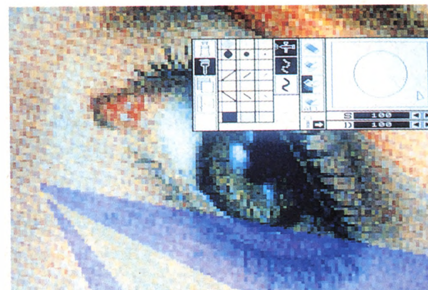
綺麗なサンプルとオートデモあり。

まだまだ，先はあるのである

X68000のグラフィックツールといえば，多くのユーザーや開発者がZ'sSTAFFを基準にしてきた。それはそれでいいとして，Z'sSTAFFが完璧なソフトか？　というと，決してそんなことはないのである。そのひとつの例をマジックパレットが証明したわけだが，まだまだいろいろ便利な機能はあるはずである。マジックパレットだって，早く次のバージョンを！　てな感じだ。画像取り込みの柔軟さと，任意矩形の変形はほしいところ。グラデーションなんて簡単に綺麗な効果が出せるだけで乱用すると見苦しいだけだし。

えっと，コンピュータを使って絵を描くことの意義を考えてもらいたい。絵心のある人がペンをマウスに持ち替えて，ああ，よかったね，という時代は過ぎ去った。わざわざマウスを持たせるのだから，結局ペンで絵の描ける人でないと使いこなせない，というのは変である。Z'sTRIPHONYやC-TRACEなどはデッサン力がなくても，センスと根性と待つだけの暇とちょっとした頭があれば誰でも使えるものだった。2Dグラフィックツールも，そんなものが出てきてもいいではないか。

たとえば，遠近法矩形や，始点と終点で太さの変えられるペン。任意の方向へのグラデーション。多彩なアンドゥや下書きプレーン（メモリの関係で大変だろうけど）。別に65536色でなくとも，32768色でも16384色でもいいので，そういった支援機能を充



メニューは邪魔にならないポップアップ式

たかがツールされどツール

CGコンテストの審査などでよくいわれることですが，応募されてくるものに「こんな機能を使ってみました」みたいな作品が結構多いのです。ツールの豊富な機能を使うのはいいのです。でも，それに振り回されて自分の表現したいものがあやふやになっては駄目ですね。作品はツールの機能紹介ではないのですから，饒舌すぎないオリジナリティのある作品を描いてもらいたいのです。

それには自分にあったツールを深すことも必要でしょうし，最終的には自分自身で組んだ，自分のためのお絵描きツールを使うのがベストなんでしょうね。

パソコン通信を始めてから，PC-9801で描いたイラストを見る機会が非常に多くなりました。うまい人の絵を見ていると，16色という限定された色数を巧みに利用してとても美しい効果を表現しています。レイトレーシングや取り込み画像など特殊な用途以外なら，多色よりむしろ少色のほうがセンスのある色彩設計ができるのではないのでしょうか。

ところで16色768×512モードのCGツールってありませんねぇ。あればPC-9801の絵を利用しやすくなるんですけど。どっかでエスキースみたいな16色CGツール出しませんかねぇ。やっぱ自分で作るしかないのかなぁ。　(T.F.)

実させるのも手だろう。だいたいにして，1万色あればたいていこと足りるはずだ。
あと，いろいろと難しいだろうけれど，PICファイルのサポートもあると助かる。
それでもって，一番ほしいのが，キーボードマクロと自動実行マクロと数値関数（三角関数など2次曲線の描けるもの）だ。
たとえば，規則的な図形をいくつもずらして描きたい，とか，ちょっと三角関数を使った線がほしい，とか，さっき描いたやつをもう一度描きたいなんてときはあるはずだ。マウスで行った一定の動作を覚えておいて，任意の点からそれを行えるというのがキーボード（？）マクロ。メニューから関数を選び，パラメータや軸の単位を与えて，マウスで指定した範囲に指定した色で指定した太さの曲線を描いてくれるのが関数機能。それでもって，プログラムウィンドウが開いて，ちょこちょこと簡単なプログラムを組むと，それを実行して図形を描いてくれるマクロ。
つまり，BASICでちょこちょこと描ける程度のものをグラフィックツール上でやれたら面白いだろうな，と，思うわけだ。ついでに画面に適当に描いた自由曲線をフーリエ級数展開して三角関数の組み合わせに直してくれる機能，なんてのはあったら楽しいけど，そこまではいうまい。

*　　*　　*

Z'sSTAFFを買ったはいいけれど，白い画面を前にして，グラデーションの空を描いたまま石になってしまった人や，マジックパレットを買ったはいいけれど，グラデ球をたくさん描いたまま凍ってしまった人も多いと思う。ときには石になって自分の才能に謙虚になるのもいいけれど，そうでない気楽なグラフィックだって実現できるはずなのである。
いま，思ったのだが，ドローイング系のグラフィックツール（パーツなんかを組み合わせて絵を作るツール）がない。2次元のグラフィックツールにはドローイング系のツールとペイント系のツールがあって，ここで紹介したのは全部ペイント系のツールだ。どうしてだろう。今度よく考えてみることにしよう。

●Z'sSTAFF PRO-68K ［Ver.2.0］　58,000円
ツァイト　☎03(299)0461
●マジックパレット　19,800円
ミュージカル・プラン　☎03(401)2751
●G68K version Ⅱ-PRO　22,000円
SYSTEM HOUSE OH!　☎075(502)2972
●PRISM 68K　38,000円
ウルフ・チーム　☎03(5273)4795
（価格はすべて税別）

機能比較一覧表

		Z'sSTAFF PRO-68K	マジックパレット	G68K version Ⅱ-PRO	PRISM
画面モード	メイン対応	512×512,65536	512×512×256	512×512,65536	512×512,256 512×512 256×512 512×256 256×256 (2,4,8,16,64,256,65536色)
ファイル形式	ZIM	○		○（非圧縮）	○(非圧縮)
	GL3		○	○	—
	独自		○		○
カラー	グラデーション	縦/横	縦/横/円	—	—
	スポイト	○（2種類）	○	○	○
	タイル	○	○	○	△
	トーン	○	△	○	—
	混ぜ合わせ	○	—	—	—
	濃淡	○	—	○	—
	透明色機能	—	○	—	○
ペン	太さ	19種類	7種類	24種類	19種類
	ペン先編集	○	△	—	—
	BOX/FILL	○/○	○/○	○/○	○/○
	円/FILL	○/—	○/○	○/○	—/—
	楕円/FILL	○/—	○/○	○/○	—/—
	扇/FILL	○/○	—/—	○/○	—/—
	閉曲線PAINT	○	○	○	—/—
	直線	○	○	○	○
	スプライン	○	—	—	—
	マスク	○	—	○	—
	ブラシ	○	○	○	○
	ブラシ編集	○	○	○	—
編集	ルーペ(×2)	○	○	○	○
	ルーペ(×4)	○	○	○	○
	ルーペ(×8)	○	○	○	○
	ルーペ(×16)	○	○	—	○
	矩形COPY/MOVE	○/○	○/—	○/○	○
	閉曲線COPY/MOVE	○/○	○/—	○/○	—/—
	矩形変形	○	△（画面回転のみ）	回転のみ	—
	拡大/縮小	○	△（全画面のみ）	○	○
	上下反転	○	△（全画面のみ）	—（回転で可）	○
	左右反転	○	△（全画面のみ）	—	○
	シフト	—	△（全画面のみ）	—	○
	カラーチェンジ	○	○	○	—
	パレット編集	○	○	○	○
	モザイク	○	—	—	—
	ぼかし	○	—	—	—
文字	16ドット	○	○	—	—
	24ドット	○	○	○	—
	アウトライン	○	—	—	—
	斜体	○	○	—	—
	グラデーション	○	—	—	—
	影	○	○	—	—
	縁取り	○	○	—	—
外部入力	COLOR IMAGE UNIT	○	○	○	—
	IMAGE SCANNER	○	○	○	—
座標表示		○	○	○	○
方眼紙		○	○	—	○
アンドゥ		○（要2MB）	△	○	—
画面数		1	3	1	2
スプライト・セーブ		—	○	—	○
アニメーション		—	—	—	○
子プロセス		—	○	—	—
オートデモ		—	○	○	—
おまけ		ZIMLOAD他	BASIC関数	ZIM↔GL3変換	

ギザギザのないグラフィック関数

アンチエリアスとは?

コンピュータグラフィックでの強敵のひとつジャギー。今回はこれを追放すべく，新しいグラフィック関数を揃えてみました。もっとエレガントなラインルーチンと高機能なペイントルーチンなどによる高画質な2Dグラフィックワールドを構築していきます。

Tan Akihiko 丹　明彦

というわけで，2次元グラフィックである。これまでは3次元グラフィックが主だったので，次元がひとつ落ちたことになるのだが，それはつまり，質的にも一段と落ちたことなのか?　いやいやとんでもない。2次元のほうが3次元よりもずっと身近で表現しやすいのである。そして表現しやすいぶん，人は精魂こめて絵を作り上げるし，質的にも高いものができる。そのことはOh!X誌に毎日のように送られてくるイラストを見てもわかる。とにかく，層の厚さが違うぶん，競争も激しいし，いいものしか残らない。これはとてもいいことである。

さて今[redacted]目標は

これから紹介するのは，コンピュータのスクリーン上によりよい2次元の1枚絵を作るための道具である。といってもX-BASICのグラフィック関数とやっていることは基本的には同じ。1つひとつの関数の動作は非常にプリミティブなもので，現段階では「ペンと紙とスクリーントーンをキーボードに持ち換えた」のと同じような感覚で使うことは，残念ながらまず無理である。優れたグラフィックツールであるZ'sSTAFFでさえ，ただペンをマウスに持ち換えただけなのとは少し違うのだが，それとは次元が違う。

今回制作しようというX-BASICの外部関数は，マウスから入力するといったユーザーインタフェイスについては無視である。つまりその部分はユーザーであるあなたにお任せ，ということになる。用意したのはやや強力なラインやペイントなのだから，それをあなたがどう活用しようとまったく自由である。

X68000でラインやペイントを使った2次元グラフィックで良質なイラストを作ろうというのが今回の試みだといったが，こういう反論もあるだろう。「X-BASICにだってラインやペイントはあるぞ，どうしていまさら作り直す必要がある?」と。その考えは甘い。X68000の標準グラフィック関数は，せっかくの65536色を生かしきっていないのである。

コンピュータで描いたイラストの多くがどうして雑誌の表紙を飾りうるだけのクオリティを持ちえないのか。よくできてはいるけどどこか違和感のあるイラスト。そのひとつの解答がここにある。輪郭に出てくる見苦しい階段，すなわち「ジャギー」である。

'90年のトレ[redacted]ドはドッター

その昔，人間デジタイザと呼ばれる人々がいて，変な奴と思われながらも尊敬を集めていた。かれらの道具はラインとペイントであった。当時はマウスなどという便利な道具は庶民の持つべからざるものであった。Z'sSTAFFのような操作が簡単なうえに強力な表現力を持ったグラフィックツールに至っては，夢のまた夢であった。

そこで彼らは方眼紙に下絵を描き，座標値を丹念に取りながらぽちぽちとキーボードから打ち込んでいたのであった。そしてラインで線を引き，中をペイントで塗りつぶす。

いまでこそ総天然色（ちと古いか）は常識でも，8色が主流であった時代のこと，微妙な筆づかいなどは表現しえようはずもない。そのため古来の名画を模写するような試みはあまりなく，彼らの興味はもっぱらアニメ絵に向いていた。パソコン使いとアニメファンの深い関係はこうしてできあがったのであろうか。

そして時代はアナログに向かい，高品質の絵を誰でも作れる，そんな期待を感じさせるマシンの登場を見た。X68000である。ところがその期待はまだ期待の域を出てはいないのかもしれない。

X-BASICでline()関数を使ってみた方は，およそ滑らかさがないのに驚かれたのではないかと思う。もちろん，従来機種ではそれが当たり前のことだったのだが，せめて65536色モードのときくらい，もっと目に優しい線がほしい……というわけでZ'sSTAFFに期待がかかるわけであるが，こちらでも残念ながら完全なサポートはなかった。

この件の解決法はいくつかある。

・中心部が濃く周りが薄いペンを指定して，ふつうに線を描く
・ただの線を描いてあとからぼかす
・あきらめる

3番目は問題外として，どれも自然で滑らかな線にはならない。さらに共通の欠点もある。これらの方法でそれらしく見えるように線を描けたとしよう。すると，こういう線で囲んだ内側をペイントしようとしてずっこけることになるのである。ペイントできないのである。いや，できることはできるがきちんと隅々まで塗りつぶしてくれないのである（手元にZ'sSTAFFのある方はお試しいただきたい）。というわけで最後の手段として，

・ルーペで拡大して1点1点描く

ということになるのである。

現在あちこちで（市販ゲームのビジュアルシーンなどで）見かける比較的良質な画像のほとんどは，こうやって描かれている。現在のデジタルペインティング界を支えているのは，このドッターたちなのだ。

僕はこのルーペでドット打ちという作業を自分ではしたことがないので，はなはだ無責任な意見ではあるのだが，どう見ても非人間的な作業としか思えない。この点，人間デジタイザと似通っている。

しかし描いている本人は決してそうは思っていないであろう。この手の作業は慣れると苦しくはなくなるものである。それにつれて質も上がってくる。しかしどうしても職人芸になりがちである。いきおい選ばれし者の技術になってしまう。そして一般ユーザーからは変な奴とか閑人とかのレッテルを貼られてしまうのである。合掌。

今回はそこまでの質を追求するつもりはない。BASICから手軽に使えればよい。い

ろいろと遊べたらなおよい。そんな気持ちで作ってみた。

アンチエ[redacted]アシング

で，さきほどもちらっと出てきた「ジャギー」である。これは昔から再三いっているように，有限個しかないグラフィック画面のドットで，無限といってもいい細かさの画像を表現しようという要求のなかで，起こりうるべくして起こる問題である。サンプリング理論の言葉で「エリアシング」という。

これを防ぐためには，視力の限界を超えた高い解像度のCRTを使うのが完璧な解決法であろう。しかしそんなものはないし，あっても化け物のように高価であろう。

ではどうするのか。うまいことフィルタをかけて，不連続に変化しそうなピクセルの輝度の変化を補間するというのが現在もっとも効果を上げている方法である。

黒い線を引いたつもりでも，その縁の部分には微妙に灰色のピクセルが並んでいて，遠くから見れば滑らかな線に見えるのである。境界をぼかしてごまかしているのと混同されがちだが，これはぼかし処理とはまったく異なるもので，アンチエリアシングと呼ばれる。

百聞は一見にしかずというわけで，まずはなにもいわずにリスト1を実行していただきたい。いうまでもなくX-BASICのリストである。

線が4本画面に見える。そのうちいちばん上といちばん下の線は画面の中央でがたんと1ドット上がり，2本の線を1ドットずらして継ぎ合わせたような印象である。1本の線には見えない。

対して内側の2本はわりと綺麗な線に見える。そしてこの4本は，全体としては平行線である。右端と左端を見ると，確かに等間隔である。とすると，内側の2本はまっすぐな線に見えるようだが，状況から見て，どこかで1ドット上がっていなくては辻褄があわない。

ここでZ'sSTAFFをお持ちの方は，ルーペを使って，内側の2本がどのような色使いをしているのかを見れば，アンチエリアシングの原理がおぼろげにでもわかることであろう。

しかしこれではあまりにも応用がきかない。今回作ったのは，もっといろいろな線にも使えるようなプログラムである。その具体的なアルゴリズムの説明はあと回しとしよう。とりあえず使えることが大切だ。

使い[redacted]ある

どういう形式で実現するか迷ったのだが，手軽に使えるX-BASIC外部関数という線に落ち着いた。その関数本体はC言語で書いている。

X-BASICの外部関数をCで書くときの注意や，コンパイルの手順などは囲みにしてあるのでそちらをご覧いただくとして，いま，あなたの手元には今回作った外部関数anti.fncを組み込んだX-BASICが起動しているものと思って話を続けることにする。

関数のリファレンスマニュアルを表1に掲げる。anti.fncにはこの表にない関数も収録してあるが，隠し関数のようなものなので，とりあえずは表1に載っているものだけを使っていただきたい。

anti.fncを使いこなすには，表1や図1にも出てきている「点列」というデータ構造の把握が不可欠である。というよりもそれがほとんどすべてである（点列にはCのソースファイル中ではPTSという型を与えている。pointsを略して命名した）。

点列の基本単位は整数3個で，それが（頂点の数＋1）個並ぶかたちになる。X-BASIC上では，

int pts(10, 2)

と宣言する。BASICの配列の宣言は，Cのそれと少し違っていて，同じ宣言をCでは，

int pts[11][3];

リスト1

```
10 /* アンチエリアシングの原理
20 /* 1ドットの段差を持つ線
30 screen 1,3,1,1
40 /* アンチエリアシング
50 for x=0 to 511
60   i=32*x/512
70   pset( x,199,rgb(i,i,i) )
80   pset( x,200,rgb(31-i,31-i,31-i) )
90   pset( x,204,rgb(i,i,i) )
100  pset( x,205,rgb(31-i,31-i,31-i) )
110 next
120 /* ノンアンチエリアシング
130 line( 0,210,511,209,65534 )
140 line( 0,195,511,194,65534 )
```

図1　点列のデータ構造

dim int pts(N, 2)で定義する。Nは点列のおおよそのサイズ。

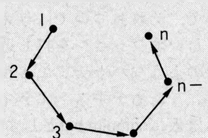
i＼j	0	1	2	
0	n	0/1	0/8	ヘッダ
1	x_1	y_1	w_1	頂点の情報
2	x_2	y_2	w_2	
3	x_3	y_3	w_3	
⋮	⋮	⋮	⋮	
n−1	x_{n-1}	y_{n-1}	w_{n-1}	
n	x_n	y_n	w_n	
n+1				未使用
⋮	⋮	⋮	⋮	
N				

pts(i, j)の意味

〔ヘッダについて〕(i＝0)

pts(0, 0)＝n　　頂点数(≦N)

pts(0, 1)＝0または1　　点列のタイプ



pts(0, 1)＝0　多角形は開いている

pts(0, 1)＝1　多角形は閉じている

pts(0, 2)＝0または8　オーバーサンプリング座標のフラグ

pts(0, 2)＝0　オーバーサンプリング処理がされていない

pts(0, 2)＝1　オーバーサンプリング処理ずみ

〔頂点について〕(1≦i≦n)

(x_i, y_i)　第i点の座標

w_i　第i点での線の太さ(aa_lines( )関数)

とする。BASICは添え字の最大値を，Cは1次元あたりの要素の数を基本にしているからだが，あとの参照や代入のしかたは両者ではほとんど変わらない。

点列の宣言は前述のとおり2次元配列で行うが，第1添え字(10)は頂点の数の最大値というか，その目安を適当に決めて設定する。たとえば複雑な形なら値を大きくする。曲線を記録する(後述)ときも大きくする。第2添え字のほうは2に固定である。

点列の構造について少し解説しよう（図1）。頭から3要素，pts(0,j)は少し特殊で，ヘッダと呼んでいる。pts (0,0) には実際の頂点数が，pts (0,1) には点列のタイプが，pts (0,2) にはオーバーサンプリング倍数が入る。

点列のタイプは2つに分かれる。それを理解する助けとして，1本の紐を想像してもらいたい。その紐が点列を表している。いま，その紐の端と端を結んだとする。その状態が，点列タイプ＝1の状態で，循環していると名づける。要するに閉じているわけである。そうでない，開いている状態が点列タイプ＝0というわけである。

オーバーサンプリングについては，もう少し後ろで説明するが，予備知識として簡単にいっておくと，今回の目玉であるアンチエリアシングに使う技法である。

ひとつのピクセルをより細かく分割して図形を描き，出力する段階で平均すれば，最終的に出てくる図形の輪郭が滑らかになるという思想に基づいている。座標系を，ピクセルのサンプリング周波数よりもっと細かく取るから，オーバーサンプリングと呼んでいる。今回は8倍オーバーサンプリングとしたので，pts (0,2) には0か8が入る。

今回のプログラムの作り方

X-BASICの外部関数をC言語で作るわけであるが，今回のプログラムは，
・内蔵の関数（機能）が比較的多い
・それぞれの処理が多少複雑
・したがってプログラムサイズが大きい
・たったひとつの関数をデバッグするのにいちいち全部コンパイルしなおしていてはやりきれない
というわけで，分割民営化，じゃない，
　　分割コンパイル
の採用に踏み切った。複数のソースファイルを別々にコンパイルして，最後にリンカを使ってひとつにまとめるやり方のことである。僕も今回ほどバラバラにしたのは初めてだが，いざやってみると非常に快適である。

0)　環境

最初に開発環境を確認しておこう。

使うCコンパイラはXCかGCC。コンパイラはどこに置いておいてもいいが，パスは通っていなくてはならない。コンパイラのほかにもアセンブラ（as.x）とリンカ（lk.x）が必要である。これらにもパスを通しておくこと。当然ながらテキストエディタも必要。僕はmicroEmacsを使っているが，標準的なのはed.xであろう。

設定しておかなくてはならない環境変数もいくつかある。autoexec.batなどに次の設定がされているかどうか確認しておくこと。システムがAドライブでRAMディスクがFドライブの場合，

```
TEMP F:
SET lib=A:¥LIB
SET include=A:¥INCLUDE
```

BASICの入っているディレクトリは，

```
A:BASIC2¥
```

とする。そうでない方は各自のシステムにあわせて読み換えていってほしい。

ほかに大切なのはインクルードファイル（＊.h）およびCのライブラリ（＊lib.a）である。それぞれ，

```
A:¥INCLUDE¥
A:¥LIB¥
```

に収めておくこと。C compiler PRO-68Kのシステムディスクの設定なら基本的には安心してよい。そうそう，GCCの場合は，gnulib.aというライブラリもあるが，これもA:¥LIB¥に収めておけばよい。

1)　ソースリスト作成

環境設定ができたら，さっそくソースリストを作ろう。打ち込むリストは次のとおり。すべてふつうにテキストエディタで打ち込む。

・anti.s　　（外部関数ヘッダ）
・anti.h　　（マクロ定義ファイル）
・main.c　　（引数リスト宣言）
・pts_curve.c　（自由曲線の発生）
・pts_procs.c　（輪郭の処理）
・aa_lines.c　（輪郭描画）
・aa_scanconv.c（多角形塗りつぶし）
・aa_paint.c　（閉領域ペイント）
・aa_procs.c　（タイル・トーン処理）

一度に全部打ち込むのもおっくうなので，テストしながら作業を進めたい方や，必要ない関数を打ち込みたくない方は，そういう関数の名前だけ書いて中身を書かない（return（0）だけは入れておいたほうが安全だが）という手が使えるので参考にしていただきたい。

2)　コンパイルおよびアセンブル

ソースリストを打ち込んだら，それぞれをコンパイルする。ただし，anti.sとanti.hは例外。anti.sはアセンブラ（as.x）でアセンブルする。

```
as /u anti.s
```

エラーがなければ，anti.oというファイルができる。anti.hのほうはただのインクルードファイル（それぞれのCのソースにインクルードして使う）で，それ自身を単独でコンパイルする必要はない（してもなにもできてこない）。

さて，Cのプログラムのコンパイルだが，こちらもふつうどおりではない。分割コンパイルなので，リンクフェイズまで一気に突っ走ってはいけない。～.oの段階で止め，最後にリンクするのが分割コンパイルである。だからリンクフェイズの直前でコンパイルをやめるスイッチをコンパイラに与えてやらなくてはならない。これがXCとGCCでは違っていて，それぞれ，

```
cc /L ～.c
gcc -c ～.c
```

である。また，GCCの場合は最適化オプションが豊富なので，それもついでに与えよう。いちいち長たらしいオプションを打ち込むのは面倒なので，次のようなバッチファイルを作ることをすすめる。これもテキストエディタで書く。ファイル名は仮にcompile.batとしよう。

(XCの場合)

```
cc /L %1.c
```

(GCCの場合)

```
gcc -c -O -fstrength-reduce -fomit-frame-pointer -finline-functions %1.c
```

バッチファイルができたなら，

```
compile main
compile pts_curve
    ⋮
compile aa_procs
```

と各ソースごとにコマンドラインから実行する。もしエラーが発生したりバグを取ったりしたファイルがあれば，そのファイルだけをコンパイルしなおせばよい。

3)　リンク

ここまで無事終了したら，～.oというファイルが8つできていることであろう。そこで仕上げのリンクフェイズ。

```
lk /o anti.fnc anti.o main.o pts_curve.o pts_procs.o aa_lines.o aa_scanconv.o aa_paint.o aa_procs.o %lib%¥clib.a (%lib%¥gnulib.a) %lib%baslib.a
```

カッコ内のgnulib.aというのははGCC専用のライブラリで，いうまでもなくXCでコンパイルする人には必要ない。/oオプションを使って，ふつうならanti.xとなる出力ファイルの名前を外部関数の名前anti.fncにする。実はX-BASICの外部関数の正体は実行形式ファイルと同じである。ただ名前がそうなっていないだけ。

4)　インストール

あとはX-BASICにできたてのanti.fncを組み込むだけである。まずBASICのディレクトリにanti.fncを転送する。

```
copy anti.fnc A:¥BASIC2¥
```

それからBASICのディレクトリ上のコンフィギュレーションファイルをテキストエディタで書き換える。標準ではbasic.cnfというファイル名である（X-BASICは/cオプションを使って指定したコンフィギュレーションファイルで立ち上げることもできる）。

以下はその一例である。大切なのは最後の1行。

```
FREE = 128
WIDTH = 64
BEEP = ON
CAPS = OFF
FUNC = GRAPH
FUNC = PIC
FUNC = ANTI
```

ほかにも音楽関係の外部関数を組み込んでおけば，音楽を演奏しながら絵を描くという芸当もできるだろう（してなんになる）。ところで下から2行目のpic.fncというのは，やはり本誌6月号の付録ディスクについていたPIC形式の画像ファイルをセーブ/ロードする外部関数。描画の遅いanti.fncにとってはとてもありがたい相棒である。

これでやっと使えるところまでこぎつけた。正直いって，Cとアセンブラを扱いなれた人にはこんな説明は退屈なだけかもしれない。

pts(i,j)は，$1 \leqq i \leqq$ pts(0,0) である i については i 番目の頂点の情報を格納する。pts(i,0)にはx座標が，pts(i,1)にはy座標が，pts(i,2)には線の幅がそれぞれ入る。

それでは動作チェックも兼ねて簡単な使い方を練習しよう。まずは点列の宣言の方法から。

例1） V字型

```
dim int p1(3,2) = { 3,0,0
,100,100,8
,200,300,8
,300,100,8 }
```

例2） 三角形

```
dim int p2(3,2) = { 3,1,0
,100,100,8
,200,300,8
,300,100,8 }
```

この2つのサンプルのあいだでは，点列タイプ（pts(0,1)）だけが違うことに注意しよう。

点列の定義ができたら，それを使ってなにか描いてみよう。その前に，完全に制作者（要するに僕）の都合なのであるが，点列をオーバーサンプリング座標系に変換しなくてはならない。変換をかけておかないと，この先出てくるほとんどの関数が使えない。ま，ここはおまじないとでも思っておこう。

```
pts_oversample(p1)
pts_oversample(p2)
```

次に，画面モードを65536色モードに変える。ちょっと手抜きなことに，描画関数の中に画面モードのチェックを入れていないので，忘れずに実行しておくこと。

```
screen 1,3,1,1
```

では先ほど作った三角形を画面に出してみよう。

```
aa_lines(p2, rgb (31,31,31))
```

なかなかダルいが，おしまいまで待とう。白い三角形が出てくると思う。

お次はいまの三角形の頂点を通る曲線を作ってみよう。それにはまず，曲線を格納する配列をひとつ用意する。というのも，曲線は短い線分をたくさんつなげてそれらしく見せるようにしているからだ。そのため，ある程度多くの頂点も記録できるように大きな配列を用意する。余裕を持って，

```
dim int p3(1000,2)
```

と大きめに宣言しておき，すかさず，

```
pts_curve(p2,10,10,p3)
```

を実行。

pts_curve()は曲線を生成するだけの関数なので，画面にはなにも出ないはずだ。ちょっとしたら戻ってくるので，できた曲線を見てみよう。さっきと同様に，

```
aa_lines (p3,rgb (31,0,0))
```

今度は赤い色で三角形のカドを取ったような曲線が出てくるはず。

さてここでいったんご破算願おう。

```
wipe()
```

そして新しい気持ちでもっと妙な形を試してみることにする。

```
dim int p4 (6,2) = { 6,1,0
,100,100,8
,200,300,8
,300,100,8
,400,400,8
,300,200,8
,200,400,8 }
```

例によってオーバーサンプリング座標に変換するおまじない。

```
pts_oversample(p4)
```

この「N」をひっくり返したような多角形の頂点を通る曲線を作る。

```
dim int p5(2000,2)
pts_curve(p4,8,8,p5)
```

さっきは輪郭線だけだったが，今度はこの曲線の内側も塗りつぶしながら描く。

```
aa_scanconv (p5,0,65534,0,0)
```

白い変な形が現れる。その中を赤でペイントしてみよう。

```
aa_paint(250,250,0,rgb(31,0,0),0,0)
```

ちなみに，このaa_paint () の代わりに，

```
paint (250,250,rgb(31,0,0))
```

を実行してみると，aa_paint()がアンチエリアシングに対応しているありがたいペイント関数であることがわかることだろう。

以上の動作に支障がなければ，ほぼバグはないと考えていいだろう。表1の関数リファレンスを参照しながら，上の例題の数値をあちこちいじって実行してみよう。そして，それぞれの関数がどういう機能を持ち，どんなパラメータを与えるとどんな動作をするか，そういうことを理解して，さらに難しい作品へと進んでいってほしい。

アルゴリズム解説

ソースリストが思ったより大きくなってしまい，我ながら驚いている。こんなものの説明をすることを考えるだけで胸焼けである。ま，すべてはソースリストが語ってくれるということで，コーディングするうえでの細かい注意は，ソースリストに入れたコメントに頼ることにし，ここではアルゴリズムの心を語ることにする。

今回の外部関数を構成するための主要なアルゴリズムはいくつかある。幸いなことに，過去のOh!X誌ですでに僕が紹介しているものも多いので，適宜参照していただきたい。



アンチエリアシングの奇跡

オーバーサンプリング

アンチエリアシング技法のなかでももっともポピュラーな方法のひとつが，このオーバーサンプリングである。レイトレーシングやZバッファといった3次元CG技術をアンチエリアシング対応にする場合，必ずといっていいほど用いられるのもオーバーサンプリング。

ここまでの説明でもちらっと触れているのだが，まず事実として，ピクセルのサンプリングレート（要するに解像度）はかなり高いように見えて，人間の目をごまかしおおせるほどには高くないということがある。そこで多色表示の利点を生かすことが考えられた。ともすれば急激で不連続的になりがちなピクセルの輝度変化をもっと滑らかにし，曖昧な（少し語弊があるが）輪郭を作れれば，目に優しい画像ができあがる。

そのために，いったんピクセルよりも高いサンプリングレートで画像を生成しておく。このときの最小の処理単位は，ピクセルよりもさらに小さな画素であり，サブピクセルと呼ばれる。

ちなみに1本のスキャンラインも数本のさらに細いスキャンラインに細分されることになり，サブスキャンラインと呼ばれる。今回の外部関数では8倍オーバーサンプリングを採用している。この場合1ピクセルは8×8＝64サブピクセルからなる。

描画アルゴリズムは従来の（オーバーサンプリングを用いない）アルゴリズムを拡張して使う。ただ処理単位がピクセルでなくサブピクセルになっているだけである。

そして，1ピクセル中の全サブピクセルの輝度を平均してスクリーンに出力すれば，粗いピクセルにそれ以上の解像度を持たせたのと同等の効果が得られるという仕掛け

になっている。

誤解を恐れずにいうなら，アンチエリアシングは人間の目を巧みにごまかす技法であるともいえる。もちろん，ピクセルをよく見ればそんなごまかしはすぐわかってしまうし，1ピクセルを下回るような細かい図形には効果が薄くなってしまうといった欠点はあるものの，いたずらに解像度を上げるよりもずっといい方法なのである。

今回の描画アルゴリズムでは，サブピクセルの輝度を1つひとつ配列に持っておくことはしなかった。2次元なので，基本的に隠面処理など考える必要はないし(*)，それならば「いま描画しようとしている図形が各ピクセルのうちいくつのサブピクセルを占めているか」という情報だけが重要だとわかる。これをピクセルあたりの寄与率と呼ぼう。以後はαという記号を使うことにする。

8倍オーバーサンプリングの場合，サブピクセル数は0から64の値をとる。αはこれを64で割った値，つまり0≦α≦1の間の値をとる。ピクセルと図形がまったく重ならない場合はα＝0だし，ピクセルを図形が全部覆っている場合はα＝1。境界部だけでαは0でも1でもないいろいろな値をとる。

αは一般に実数だが，プログラム上は実数よりも整数のほうが取り扱いが楽なので，ひとまず0≦α’≦64で格納しておき，最後に64で割っている。これでも結局は同じである。

スクリーン出力の段階では，α合成と呼ぶ方法を用いる。背景が真っ黒な場合はαがそのまま輝度になるのだろうが，もちろんいつでもそんなことがあるはずはなく，ふつうは，適当な比で図形の色と背景の色を合成しないと，輪郭が変になってしまう。この比にαを用いるのである。つまり次の比で混合する。

図形の色：背景の色＝α：1－α

参考）　この方法の画質をもっと上げる方法として，重みつけ平均化をすることも考えられる。αを出す段階で，ピクセルの中心部のサブピクセルほどαに大きく寄与するようにプログラムを組んでおくのである。今回採ったのは単純平均化で，どのサブピクセルも同じ重みをもっていることになっている。

Bresenha[redacted]アルゴリズム

昨年解説したZバッファアルゴリズムの前フリとして線分描画を説明した（1989年7月号）。一般に線分の傾きは実数である。実数である線分の傾きを相手にしながらも，Bresenhamアルゴリズムはすべての演算を整数ですませてしまう。このアルゴリズムは，実に応用が広い。たとえば本誌5月号のグラフィック拡大縮小にも使っている。

Bresenhamアルゴリズムの核となる部分を以下に示す。(x1,y1) から(x2,y2)へ色cで線分を引く。ただしここではx1<x2,y1<y2である。ほかの場合についてもそれほど難しくない拡張で対応できる。

Bresenhamアルゴリズムの基本的な考え方は，ピクセルの中心と真の線分との上下関係を比べ，真の線分にもっとも近いピクセルを点灯していくというだけのことである。この上下関係を比べるのに，誤差と呼ぶ量eを使って処理を効率的にしている。

```
dx = ( x2-x1) ;
dy = ( y2-y1) ;
e = -dy;        （誤差の初期値）
for (x=x1,y=y1;x<=x2;x++) {
    pset ( x,y,c ) ;   （ピクセル点灯）
    e += (2*dy);
    （1ピクセルあたりの真の線分の上昇分）
    while ( e>=0 ) {
    （真の線分が上にあるあいだは）
        y++;
    （ピクセルの座標を上げる）
        e -= (2*dx);
    （その分だけ真の線分との距離を詰める）
    }
}
```

もっと詳しく知りたい方は1989年7月号の記事を参照してほしい。

[redacted]

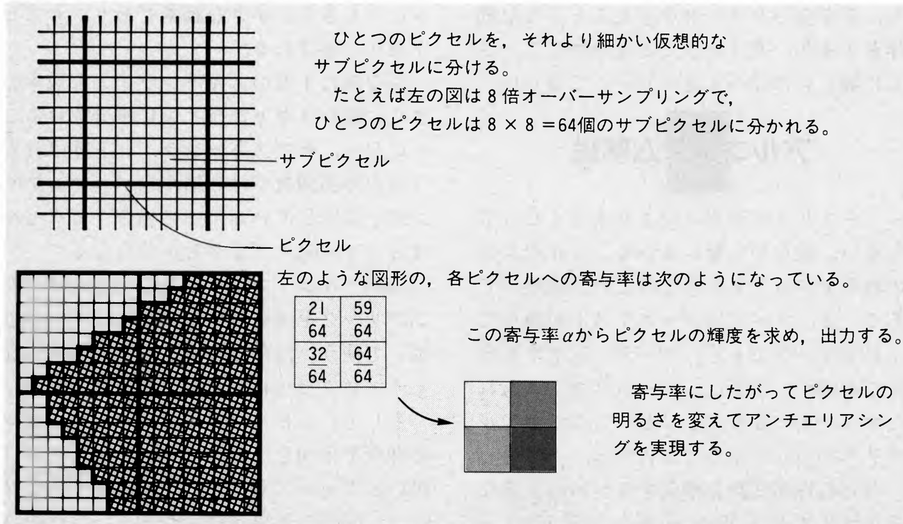
ただの線分ならば上のBresenhamアルゴリズムを使うのだが，アンチエリアシング対応となるとそう簡単にはいかない。しかも今回は欲張って，線分の幅を変えられるようにしたのでよけい厄介である。

それでは，(x1,y1)から(x2,y2)へ幅wで線分を引くことを考えよう。といってもそれほど難しいことではない。まず描きたい太い線分を1ピクセル間隔で切る。イメージとしては輪切りである。そしてそのひとつひとつ（幅1ピクセルで長さhピクセルの小線分）をスクリーンに張り付けるのである。切り口の長さhは，ピタゴラスの定理(おお懐かしい)を使って求めることができる。

ここまでわかればあとは簡単。まず太い線分の下端（これも線分になる）を通常の線分と同じようにBresenhamアルゴリズムで発生させる。

具体的には(x1,y1−h/2)と(x2,y2−h/2)を結ぶ線分，すなわち太い線分の中心からhの半分だけ下にずれた線分である。そして，この下端の線分の上に長さhの小線分を並べていけばいい。これは，まっとうに描けば傾いた長方形になるはずの太い線分を，平行四辺形で近似したことになる。あまり線分が太くないうちはたいして不都合はおきないが，太くなってくると不自然さも目立つし，ときには破綻することもあ

図2　オーバーサンプリング



（＊）3次元CGだとさすがにこんないい加減なことではすまされず，きちんとサブピクセル数だけのZバッファなりを用意し，隠面処理をきちんと終えてから合成するという手順が要求される（これはあくまで原理的な話で，実現するうえではもっと効率的な方法も提案されている）。

る。このことはあとで触れる。

いずれにせよこれで太い線分は描ける。あとはこれをオーバーサンプリング座標系で処理し，寄与率αをピクセルごとに求めてからα合成を行うように拡張するとよい。ここから先は単なる力仕事である。また，aa_lines () 関数はただ1本の線分ではなく数個〜数百個の点列を結んで連続描画を行うので，それ相応の処理も考える必要がある。

特にひとつの線分から次の線分に移るときは，前者の終点での寄与率を記憶しておいて後者の始点へとつなげていかなくてはならない。線分1本1本ごとに寄与率を初期化していたのでは，線分の継ぎ目継ぎ目でピクセルが暗くなってしまうからである（これは現実に失敗した）。

さきほどほのめかしておいた欠点を説明しよう。aa_lines () 関数では，傾きが小さいときはx方向に，傾きが大きいときはy方向に処理するようにループを組んでいる。また，太い線分といっても前述のとおり平行四辺形で近似しているだけである。

そこで次のような事態は当然予想される。幅の太い曲線を描く場合を考えよう。その傾きは最初大きくてだんだん小さくなっていく。最初はy方向で処理していたのが，ある1点を境にx方向で処理するようになる。ここで曲線は，実にみっともないことに，まるでぽきんと折れたように欠けてしまうのだ。残念ながらこれを解消するうまい方法が考え出せなかった（下手な方法なら考えられないこともない）のでそのままにしてある。で，たいへん申し訳ないが，対抗策として，

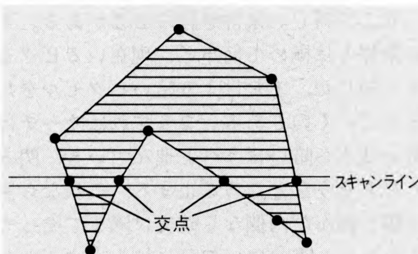
・あまり太い線分は描かせない

・太い線分を描かせる場合は，傾きをうまくコントロールして曲線が折れないように工夫する

・どうしても自由な傾きで太い曲線を描きたいのであれば，面倒でも「太い曲線の輪郭」を作り，次のスキャンコンバージョン aa_scanconv () 関数で描かせる。スキャンコンバージョンのほうはどんな曲線に対しても破綻することはない

などとしていただきたい。

図4 ソリッドスキャンコンバージョン



ソリッド領域をスキャンライン単位に細分する。スキャンラインと輪郭の交点は，Bresenhamアルゴリズムで求める。

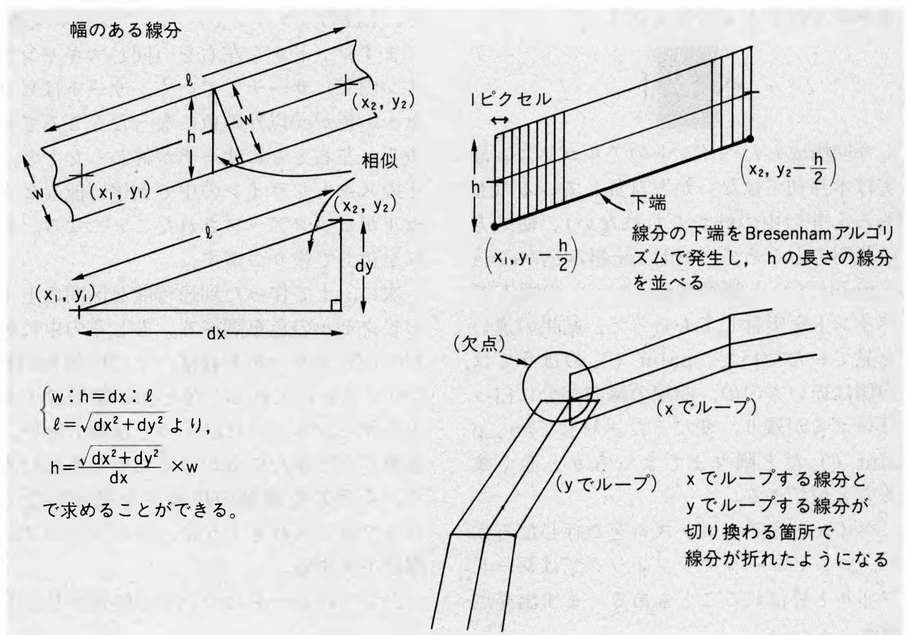
ソリッドスキ[redacted]コンバージョン

多角形を描画するもうひとつの方法で，上のaa_lines () がワイヤーフレームモデルだとしたら，こちらのaa_scanconv () はサーフェスモデルだといえるし，2次元ソリッドモデルだともいえる。要するに中身のつまった（というのも変だが）多角形を描画する。これまた1989年7月号で解説ずみである。今回のはそれのアンチエリアシングバージョンである。ただの移植ではなく，データ構造を工夫し，無駄な処理を省くなど，手を入れてある。

ソリッドスキャンコンバージョンの原理はそれほど難しくない。まず目的とする多角形をスキャンライン単位に細分する。そうするとたくさんの線分ができるので，1本1本スクリーンに張り付ければよい。

といっても抽象的というか感覚的すぎるので，もう少しアルゴリズムのほうにすり

図3 線分の描画



リスト2

```
 10 screen 1,3,1,1
 20 fill( 0,0,47,47,rgb(0,0,31) )
 30 symbol( 1,1,"色即",1,1,2,rgb(31,0,0),0 )
 40 symbol( 1,25,"是空",1,1,2,rgb(31,0,0),0 )
 50 symbol( 0,0,"色即",1,1,2,rgb(28,28,0),0 )
 60 symbol( 0,24,"是空",1,1,2,rgb(28,28,0),0 )
 70 tile_get( 0,0,0,47,47 )
 80 tone_get( 0,0,0,47,47 )
 90 fill( 0,0,47,47,rgb(16,16,16) )
100 tone_get( 1,0,0,47,47 )
110 wipe()
120 dim int p(10,2)={3,1,0
130      ,128,128,0
140      ,256,384,0
150      ,384,256,0}
160 pts_oversample( p )
170 dim int p1(10,2)={7,1,0
180      ,64,128,0
190      ,128,384,0
200      ,192,128,0
210      ,256,384,0
220      ,320,128,0
230      ,384,384,0
240      ,448,128,0}
241 dim int p2(2000,2)
242 pts_oversample( p1 )
243 pts_curve( p1,8,32,p2 )
244 whitepaper()
245 aa_lines( p2,0 )
250 /*aa_scanconv( p,1,0,0,0)
260 /*whitepaper(): aa_scanconv( p,0,rgb(31,0,0),1,0 )
270 aa_scanconv( p,1,0,1,1 )
```

寄ろう。まず多角形を細分する作業は，多角形とスキャンラインの交点の座標を求める処理に相当するが，これは輪郭をBresenhamアルゴリズムで発生すれば容易に求めることができる。またスクリーンに張り付ける作業は，求めた交点の間に線分を引く処理に相当する。やはり詳しい話は1989年7月号に譲る。

アンチエリアシング化に際しては，先ほどのaa_lines（）と同様のことをする。まずすべてオーバーサンプリング座標系で計算する。もちろんスキャンラインではなく，サブスキャンライン単位で処理をするのである。そしてピクセルごとに寄与率αを求めて背景とα合成を行う。

ペイント

今回構成したペイントのアルゴリズムは実は本邦初ではないかと自負している（もちろん井の中の蛙かもしれない）。使い方の説明のところで，anti.fnc組み込みのaa_paint（）と標準のpaint（）の両方でペイントを実行してもらって，結果の違いを見ていただいた。paint（）のほうでは，内側は赤いものの，曲線の縁の部分に白っぽいゴミが残り，変だった。対してaa_paint（）だと隅々まできっちりと赤く塗ることができる。

ペイントのアルゴリズムをご存じだろうか。コンピュータグラフィックではシードフィルと呼ばれることもある。まず出発点がある。これをシード（種）と呼ぶ。シードとなったピクセルの色c0を記憶しておく。あとは，シードと同じ色をしていて，なおかつシードから到達可能なピクセルをすべてピックアップして，目的の色c1で塗りつぶす。

「到達可能なピクセルを探す」アルゴリズムで一般的なのはFIFOバッファを使うアルゴリズムである。FIFOはファイフォと読み，先入れ先出し(First In First Out)方式でデータを格納する倉庫のようなものである。待ち行列といったり，キュー(queue)といったりする。ついでにスタックは後入れ先出し(Last In First Out)の倉庫で，LIFO（ライフォ）バッファといえる。

まずシードから左右を（同じスキャンライン内で）サーチしていく。サーチはピクセルの色がc0以外の色になったところで止める。左右ともにサーチが終わったとき，そのスキャンラインの中で到達可能なピクセルがピックアップされたことになる。それを色c1で塗りつぶす。

次に，上で作った到達可能な区間の上下のピクセルの色を調べる。もしその中に色がc0のピクセルがあれば，その座標をFIFOバッファに入れる。色がc0の領域は1本のスキャンラインにひとつとは限らない。途中でふたまたに分かれることもあるだろう。そうした領域の代表点を過不足なくバッファに入れるように，コーディングの際は工夫する。

ひとつのシードについての処理がひととおり終わったら，FIFOバッファから座標を1組取り出してきて，新しいシードにする。もしそのシードがすでに色c1で塗りつぶしてあった場合は，そのシードを捨てる。そんなことが起こるのかと不思議に思う方もいらっしゃるだろうが，たとえばドーナツのように穴のあいた領域をペイントするときは，ぐるっと回ってきた色c1の領域がぶつかるので，FIFOバッファに入れたときは色がc0でも，FIFOバッファから出すときは色が変わっているということも起こりうるのだ。衝突したときに，どちらかのシードが無効になるわけだ。

さて，シードが有効なときは，そのシードから出発して上と同じことを繰り返す。そしてFIFOバッファが空になったとき，ペイントも終わる。

以上はふつうのペイントのアルゴリズム。だがオリジナルのペイントルーチンでも，基本は同じである。やはり到達可能なピクセルをピックアップし，シードを更新しつつ色を変えていけばよい。違うのは，「到達可能」を判定する条件である。ふつうのペイントでは，シードの色と同じであることがその条件であった。しかしこれではアンチエリアシングをかけた領域には対応できない。なぜなら，アンチエリアシングは縁の色を少し暗くすることで滑らかに見せているものだから，当然縁の部分はシードと色が同じはずはないのである。したがって縁まできっちりと塗ることは不可能。

そこで新しい条件を作る必要がある。その条件とは極めて簡単で，現在いるピクセルと同じか，またはより暗いピクセルをたどっていくのである。こうすればサーチは暗いほうへ暗いほうへと進んでいき，明るくなりそうなところで止まる。これなら黒い線で囲んだ内側なら確実に隅まで塗ってくれるし，隣の白い部分にはみ出すこともない。

イメージとしては，山の頂上からペンキを流す図を想像していただきたい。ペンキは下へ下へと流れ，一番低いところで止ま

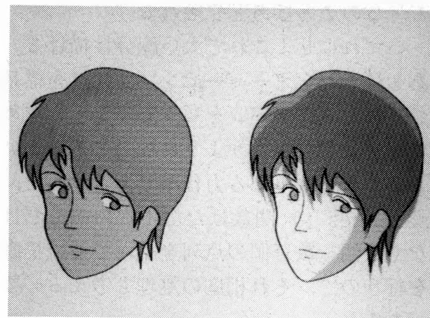
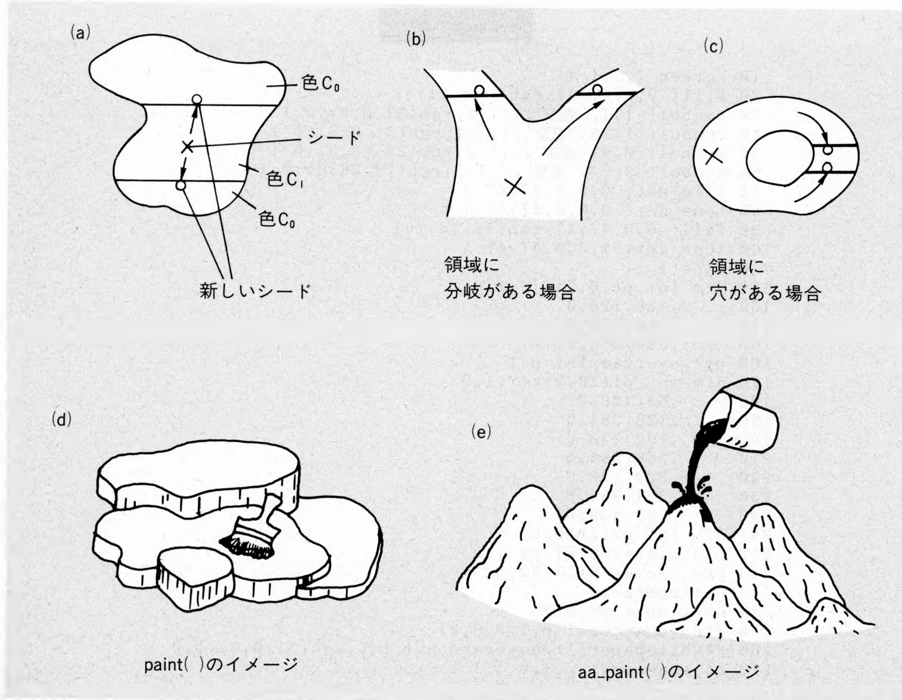

タイルも使える

図5 ペイント



る。ほかの山を上っていくようなことはしない。

注意をうながしておくが，頂上，つまり白い部分の一番明るい点にシードを置かないと，やはり正確に塗ってはくれない。

以上からもわかるとおり，オリジナルのペイントアルゴリズムでは，オーバーサンプリング座標は使わない。かわりに，ピックアップした点の輝度を寄与率αのように考えて（というよりもアンチエリアシング描画におけるピクセルの輝度はもともとαを反映したものなのだが），α合成に似たことを行う。単純に色c1で塗るのではなく，c1にαをかけた色で塗るのだ。

したがって，ペイントずみの領域を判定するのには，色がc1であるかどうか，という見分け方が使えない。ペイントずみの領域には少し暗いc1というのもある。そこでいままで出番のなかった輝度ビットをフラグとして使うことにした。X68000のカラーコードは16ビットで，上から5ビットずつ緑，赤，青の3原色が割り当てられる。そして最下位の1ビットが輝度ビットなのである。RGBと独立になっているので妙な用途に使われることが多い。Z'sSTAFFでもマスキングに用いている。

また，αの値は，aa_paint（）では赤成分の輝度を代表で持ってくることにした。これによりどんな不都合が起こるかというと，たとえば真っ青な領域は塗れないのである。赤成分がないので，全部黒と見なされるのだ。その他，明るいところから暗いところへと塗るアルゴリズムのため，暗いところから出発して明るいほうに塗っていくような塗り方もできない。

以上のように妙な制限が多いので，白地に黒く線を描いてその中を塗るという使い方をおすすめする。ついでにもうひとついっておくと，あまり細い領域を塗ろうとす

CとBASICの相性

X68000以前は，BASICの機能拡張といえば，メモリの空きエリアを捜して処理ルーチンを組み込んだり，パッチを当てたりといった，どことなく超絶技巧の香りが漂う技術であった。X-BASICでは，機能拡張を正式に許し，その仕様を公開している。さらに書こうと思えばCで書いたっていいのである。この姿勢には頭の下がる思いである。と同時にプログラマが甘やかされそうな気がしなくもない。

さて，そのX-BASICの外部関数はCで書くことができるのだが，いくつかの制限がある。

Cで素直に書けない部分について

いきなり矛盾したことをいっているようだが，BASICインタプリタと外部関数のインタフェイスを取る段階で，どうしても純粋なCだけでは無理な部分があるのである（しかし素直でなくなりさえすれば簡単に書ける。ここがCの頼もしさであり，同時に怖さでもある）。具体的には，

・外部関数のヘッダ
・引数を渡す
・戻り値を返す
・外部関数エラーのコードを返す
・エラーメッセージのアドレスを返す

部分である。

このうち戻り値に関しては，今回作った関数はみんなvoid型ということにしてしまったので問題は起きない。

それにもかかわらず，Cのソースリストでの関数の戻り値がint型（typedefを使ってFUNC型としてはあるが）なのは，Cの関数の戻り値を実はBASIC側ではエラーコードとして受け取るためである。Cの関数は（整数型の）戻り値をd0レジスタに入れてリターンするというしきたり（？）があり，またBASICのエラーコードはd0で受け取るという規則になっている。return(0)で戻れば関数が無事に終わったことを，return(1)で戻ればなにかトラブルが起きたことをBASICに知らせることができる。エラーが起きたことがわかれば，インタプリタはエラーメッセージを出し，ビープ音とともにプログラムの実行を中断してくれる。結局どちらもCで書くことができるのでこれも問題ない。

エラーメッセージが問題である。a0レジスタにエラーメッセージの先頭アドレスを入れて返さなくてはならない。これはさすがにCで書くことはできない。しかしC言語には，インラインアセンブラといって，ソース中にアセンブラのコードを直接埋め込むという技が用意されている。まっとうなCコンパイラなら必ず使えるこの技は，当然X68000上のCコンパイラ，つまり標準のXCでも本誌6月号の付録ディスクで配布したGCCでも使える。ただ両コンパイラでのインラインアセンブラの使い方は少し違っていて，XCでは，

```
  #asm
     lea __errmsg, a0
  #endasm
```

であるが，GCCでは，

```
     asm ("lea __errmsg, a0");
```

である。ここで，__errmsgはエラーメッセージを格納しているアドレスである。

GCCは本来の活動の舞台がUNIXなので，GCCの書き方はUNIX標準のCと同じである。C言語界全体を見渡せばむしろXCの作法がローカルな部類に入るのだろうが，そんなことはX68000でプログラムを作っている僕らにはなんの関係もない。どうにかして両者の違いを吸収する必要がある。

GCCのドライバはコンパイルに際して，プリプロセッサに，

```
     #define  __GCC__
```

と指定したのと同じことを自動的に行う。ま，環境変数みたいなものだ（本当は全然違う）。今回はこれを利用して条件コンパイル（#if～#else～#endif）をする方式を採った。

```
     #ifdef  __GCC__
         asm ("lea   __errmsg,a0");
     #else
     #asm
         lea   __errmsg,a0
     #endasm
     #endif
```

しかし読者の方はこんな面倒なことをする必要はない。各自の使いたいコンパイラにあわせた部分だけを打ち込めばそれでよい。

お断りしておくまでもないと思うが，GCCだけ手に入れてもコンパイルはできない。コンパイルに際してはアセンブラとリンカとXCのライブラリが必要なので，XCつまりC compiler PRO-68Kは必ず持っていなくてはならない。

ちなみにエラーメッセージであるが，グローバル変数の文字列として宣言するのがコツである。関数の外側で，文字列（char型配列）へのポインタとして，たとえば，

```
   unsigned char errmsg[ ] = "エラーだよ";
```

と宣言するとよい。こちらの変数名の頭にはアンダーバー（_）がつかないことに注意。コンパイラは，ソース中のラベル（関数名や静的変数名）にアンダーバーをひとつつけてアセンブラに渡すが，すでに述べたとおり，インラインアセンブラの中身にはいっさい手を出さないので，こんな配慮が必要である。いくらCで書けるといっても，アセンブラの知識が少しはないと，外部関数は書けないのだ。

さて順番が前後してしまったが，外部関数ヘッダである。これはもう純粋にアセンブラで書かないとしようがない。もちろんインラインアセンブラは使えるが，上述の条件コンパイルだと同じことを2回書く必要があるので，量が多いだけにユウウツである。ま，関数内で渡さなくてはならないエラーメッセージならともかく，ヘッダである。無理にCのソースリストの中に埋め込む必要もない。ヘッダは独立なファイルにした。それがanti.sである。

そしてもっともやっかいなのが引数の渡し方である。Cは引数を4バイトないしは8バイト単位でスタックに積み，関数に渡す。むろん呼ばれた関数側でも4バイト，8バイト単位で受ける。ところがX-BASICは引数を10バイト単位でスタックに積み，外部関数に渡す。10という数字はCにとってはとても半端な数字である。おかげでBASICからもらってきた引数を，Cのほうでストレートに受け取ることができなくなってしまっている。

で，これもしかたなくアセンブラで記述しなくてはならないのだろうかと思われた。ところがどっこい，Cの柔軟性をあなどってはいけない。引数リストを2バイト単位にばらせばどんな引数でも受けられるというのが鍵である，10は2で割り切れるのだから。

具体的には，まずダミー引数を用意する。その名もずばり，DUMMY型（正体はただのintだが）。その引数dummyを指すポインタ&dummyを，2バイト整数の配列par[ ]へのポインタにキャストするのである。これでどんな引数が来ても大丈夫だ。引数のアクセスについてはマクロをしこたま使ったので，それほど関数本体では苦労せずにすむだろう。しかし泥臭さには拭いがたいものがある。

引数リストの構造などは説明すると長くなるし，マクロの使い方さえ理解すれば十分だと思うのでもうこれ以上は説明しないが，もっと詳しく知りたい方は，本文の最後に掲げた参考文献をご覧いただきたい。親切かつエレガントな技法に出会えるであろう。

ると，途中でペイントが止まってしまうことがあるが，これはふつうのペイントでも状況は同じであろう。このペイントはかなりな好条件でないと働いてくれない，わがままペイントルーチンなのであった，残念ながら。

タイルとトーン

スキャンコンバージョンaa_scanconv()とペイントaa_paint()では，タイルとトーンを使うことができる。使い方はZ'sSTAFFのタイル＆トーンとほぼ同じで，描画の色にはカラーコード（単色）とタイルパターンのいずれかが選べ，トーンは使うか使わないかが選べる。

さらにスキャンコンバージョンでは，トーン指定の際に下地が透けて見えるか，それとも単に塗りつぶすのかを選ぶこともできる（ペイントは，そもそもアルゴリズム自体が下地の存在に大きく依存しているので，下地は透けて見えるのが当然なのである）。これにより，スクリーントーンを貼るのと同様の効果を狙っている。

ただし，下地が透けるモードでは，背景が黒いところにどんなトーンを張り付けてもなにも出ないので注意。スクリーントーンもペイントと同様，白地に引いた黒い線で絵を描き，その上に貼るのが基本である。

タイルやトーンのパターンの登録の手順について。まず画面に基本パターンを描いておいて，それをtile_get()関数やtone_get()関数で取り込んで登録する。トーンは例によって，取り込んだパターンのうち，赤成分だけを見ている。まあモノトーンで描いておけば安心。また，通常のカラーコードでは明るい（白に近い）色のほうが値が大きいが，トーン登録に限っては，暗い（黒に近い）ほうがトーンの色が濃いとみなされる。これはZ'sSTAFFをまねたのだが，こちらのほうがわかりやすいようだ。

パターンを登録しておけば，ペイントでもスキャンコンバージョンでも，タイル番号やトーン番号を指定すれば呼び出すことができる。いろいろ指定して使い方を覚えていただきたい。

登録は本番の描画に先立ってやっておいたほうが，画面が乱れずにすむ。また一度登録しておけば，BASICを終わるまでパターンは消えないことになっている(BASICの変数とは別の場所に領域を確保している)。だから標準的なタイル/トーンパターンを定義するプログラムを描画プログラムとは別に作り，BASIC起動時に一度だけrunしておけば，あとはそのパターンがずっと使える。

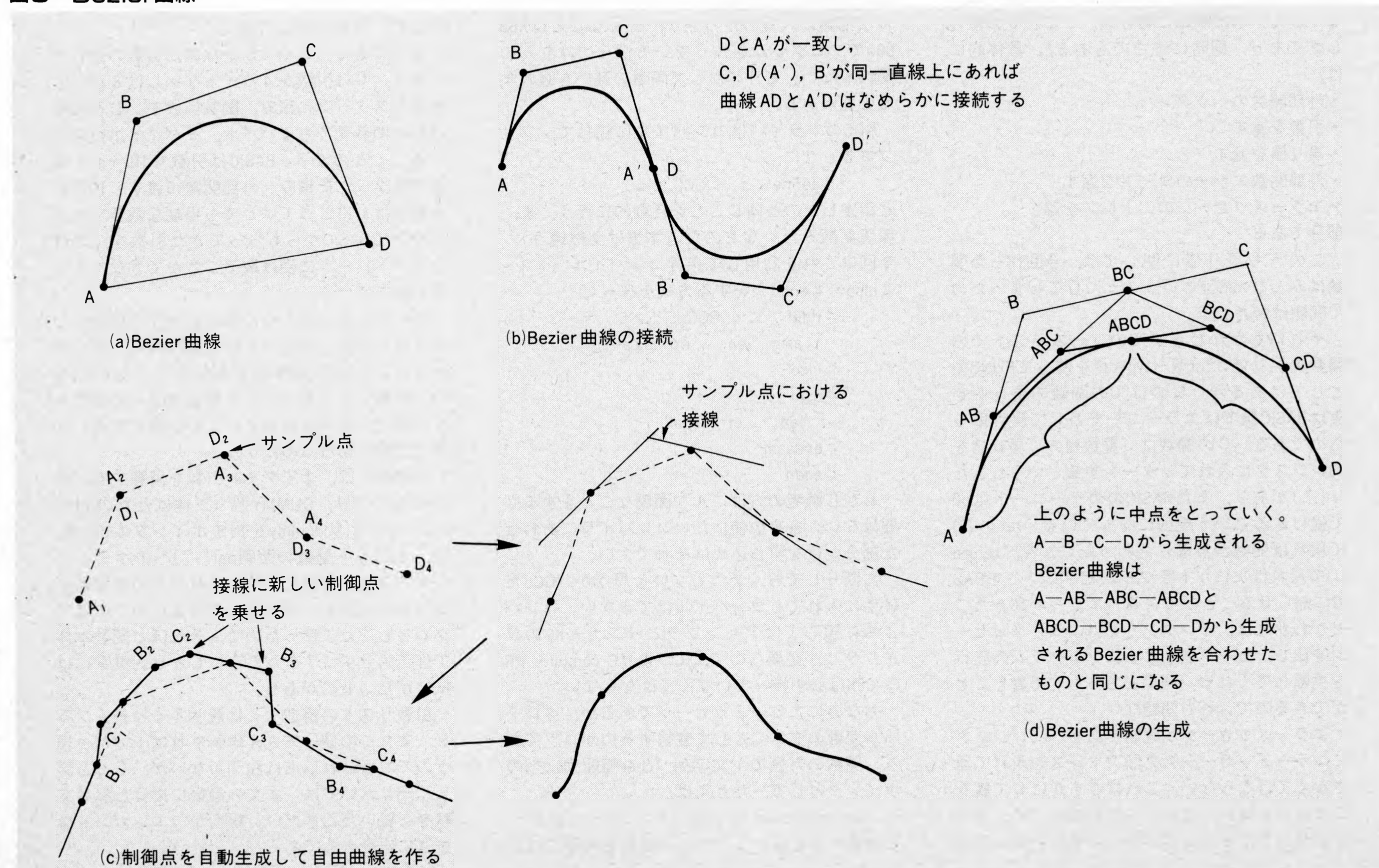
自由曲線

自由曲線にはBezier(ベジェ)曲線を使っている。Bezier曲線は4個ひと組の制御点を取る曲線である。その4つの制御点を順にA,B,C,Dとすると，Bezier曲線は次のような特徴を持つ。

・制御点Aから出発し，そこでは線分ABに接している。
・四角形ABCDの中に入る。
・制御点Dで終わり，そこでは線分CDに接している。

Bezier曲線は与えられた制御点すべてを通るわけではない（制御点BとCは通らない）が，これでは使いにくい。ふつうのユーザーなら，画面上にぽんぽんと点を置いて，その点を通る滑らかな曲線がほしい。かといっていちいち制御点BやCを手計算でつけ足していたのでは使いものにならない。下手な計算をすると，隣り合ったBezier曲線が滑らかにつながってくれないということになる。点列に記録されたすべての点を滑らかに通る曲線を生成するために

図6 Bezier曲線



は，制御点BやCをうまいこと自動的に計算して発生させる必要がある。

困っていたところで，以前に大学のコンピュータグラフィックの演習でうまいアルゴリズムを習ったことを思い出した。ここではそれを借用している。

その考え方を簡単に説明しておく。点列に記録されている点がサンプル点になる。まず各サンプル点上で，目的の曲線の接線を求める。次にその接線上に制御点BとCを乗せる。サンプル点は制御点AおよびDになる。こうしてできた制御点でBezier曲線を描かせると，曲線はサンプル点を通ってくれるし，しかもそのサンプル点上でなめらかにつながってくれる。

最後に，制御点A～Dが与えられたときのBezier曲線の発生のしかたを説明しよう。制御点の中点どうしをつないでいって新しい制御点を発生する。新しい制御点は2組できるのだが，それら発生する2つのBezier曲線をつなぎ合わせると，求めるBezier曲線が得られるのである。2分割して統合するのだから，再帰が使える。再帰的に新しい制御点を発生し，十分制御点の間隔が短くなったところで再帰を打ち切る。そんな制御点なら，いきなり線分でつないでしまっても，十分滑らかな曲線に見える。そのレベルまで再帰を続ければいいというわけ。

最後に

漫画家の道具はペンとインクと墨とスクリーントーンと，ほかになにがあるかはよく知らないが，それの真似ごとを，ある程度のクオリティでできるようにはなったと思う。それでも動作テスト用のサンプルを作ろうとしてやっぱり嫌だなと思ったのは，どうしても数値を意識しておかないとなにも作れないので，いきおいつまらない図形で我慢してしまうところである。

このままでは人間デジタイザやドッターなみの忍耐力が必要だ。マウスでぱっぱと描けるのが理想であろうが，それにはどうしてもマウスの動きに追従できるだけのレスポンスがいる。いっそアセンブラで全部書き下ろそうかと思ってしまうが，いまはコンパイラの力だけが頼りという状態だ。

ともあれ，アンチエリアシングを手動ドット打ちなど使わずに実現できる可能性は示せたと思う。ジャギーフリーのグラフィックプリミティブを装備したペインティングツールというのはまだまだ先の話であろうが，その目標への第一歩としてこの外部関数をお使いいただければ幸いである。

参考文献

(X-BASICの外部関数をCで書く方法について)

・C調言語講座PRO-68K第1回　まずはprintfより始めよ，祝 一平，Oh!X 1988年7月号，pp.98-104

・C compiler PRO-68K　プログラマーズマニュアル

・X68000 BASIC入門　最終回　必殺サンプリング戦法，中森　章，Oh!X 1988年7月号，pp.129-136

・Oh!X質問箱，村田　敏幸，Oh!X 1988年12月号，pp.129-167

(幅のある線分について)

・アルゴリズムとプログラムによるコンピュータグラフィックス［I］，S.Harrington著，郡山彬訳，マグロウヒル，pp.32-33

(ソーティングアルゴリズムについて)

・PascalとCプログラムによるアルゴリズムとデータ構造ハンドブック，G.H.Gonnet著，玄光男・荒実・松本直文共訳，啓学出版，pp.129-136

(Bezier曲線の制御点を自動生成することについて)

・コンピュータ・グラフィックスの基礎，鈴木賢次郎，長島忍，鈴木宏正，pp.A18-A20

(Bezier曲線の再帰的分割による構成法について)

・アルゴリズムとプログラムによるコンピュータグラフィックス［II］，S.Harrington著，郡山彬訳，マグロウヒル，pp.539-543

表1　関数リファレンス

オーバーサンプリング倍数はソースリスト（anti.h）のOVERSAMPLEの値を書き換えることで変えることができるが，今回は8倍オーバーサンプリングとした。

点列のフォーマット

int pts (n, 2) で宣言する。nは格納できる点列の長さの最大値。

（ヘッダ情報）

pts (0, 0) …点列の長さ，頂点数（≦n）

pts (0, 1) …点列のタイプ（0のとき片道通行，1のとき循環する）

pts (0, 2) …オーバーサンプリング倍数（点列がオーバーサンプリング座標のときには8が入る）

（1 ≦ i ≦ pts (0, 0) なる頂点 i の情報）

pts (i, 0) … x 座標

pts (i, 1) … y 座標

pts (i, 2) … 幅（オーバーサンプリング座標での値。これが8だと1ピクセル分の幅になる）

関数リファレンス

＊どの関数にも，戻り値はない。

pts_oversample (pts)

（引数）

int pts (n, 2)

（機能）

通常のサンプリングレートで記述された点列ptsをオーバーサンプリング座標に変換する。

（注意）

点列がオーバーサンプリング座標かどうかは，pts (0, 2) の値で調べる。ここにオーバーサンプリング倍数 (8) が入っていれば，その点列はオーバーサンプリング座標である。

関数のうち，点列を引数にとるものは，オーバーサンプリング座標に変換しないと使えない。

pts_curve(pts1, w1, w2, pts2)

（引数）

int pts1(n1,2), w1, w2, pts2(n2, 2)

（機能）

点列pts1の各頂点を通る自由曲線を生成し，点列pts2に格納する。その際，始点と終点での幅を　w1, w2　とし，そのあいだの線の幅を線形補間する。

（注意）

曲線を微小線分で近似するので点列pts2は多少大きめに取る。場合にもよるが，pts (1000, 2) 程度にしておけばよい。

配列の大きさが不足しているとエラーになる。

pts_append(pts1, pts2)

（引数）

int pts1(n1, 2), pts2(n2, 2)

（機能）

点列pts1とpts2をつなげ，pts1に格納する。

（注意）

pts1の最後の点とpts2の最初の点が一致するようにpts2を移動してからアペンドする。

pts1の大きさ(n1の値)は，新しい点列の長さ，つまり(pts1(0, 0)+pts2(0,0))以上用意しておくこと。

新しい点列のタイプ（片道通行か循環するか）は，もとの点列のうちpts1のタイプにあわせる。

pts_move(pts1, x, y, pts2)

（引数）

int pts1(n1, 2), x, y, pts2(n2, 2)

（機能）

点列pts1をオフセットx,yで点列pts2に移動する。

（注意）

オフセットはオーバーサンプリング座標で指定すること（各座標を8倍する)。

オフセットをx,y共に0とした場合は，点列コピーの役割も果たす。

aa_lines(pts, c)

(引数)

int pts(n, 2), c

(機能)

点列 pts に沿って，線を色 c で連続描画する。線の幅は各頂点に記録されている値を用いる。

(注意)

線の幅が太すぎると，表示が一部欠けることがある。

aa_scanconv(pts, cmode, c または n_tile, tmode, n_tone)

(引数)

int pts(n, 2), cmode, c, n_tile, tmode, n_tone

(機能)

点列 pts を輪郭とする多角形の内部を塗りつぶす。

cmode＝0 … c で指定される色（単色）で塗る。
　　　　1 … n_tileで指定されるタイルパターンで塗る。
tmode＝0 … トーンは用いない。下地の色と関係なく塗る。
　　　　1 … n_toneで指定されるトーンを用いる。下地の色と関係なく塗る。
　　　　2 … トーンは用いない。下地が透けて見える。
　　　　3 … n_toneで指定されるトーンを用いる。下地が透けて見える。

(注意)

下地は，白地に黒い線を引いた場合を想定している。

下地のうち赤成分のみを取っている（青，緑成分は無視）ので，思いどおりの結果が出ないこともある。

tmode＝2, 3の場合は，黒い背景の場所に塗ってもなにも描画しない。

aa_paint(x, y, cmode, c または n_tile, tmode, n_tone)

(引数)

int x, y, cmode, c, n_tile, tmode, n_tone

(機能)

白地の中の点 (x, y) を出発点として，黒い線で囲まれた閉領域を塗りつぶす。

aa_lines ( ) で描画されたような，境界のはっきりしない領域も塗る。

色（またはタイルパターン），トーンは aa_scanconv ( )に準ずる。

(注意)

座標(x, y)には，オーバーサンプリングでないふつうの座標を指定すること。

境界判定アルゴリズムの関係上，あまり狭い部分を塗ることはできない。

白地から黒い境界に向かって塗るので，黒地から塗ることはできない。

輝度ビットをペイントずみフラグとして用いている。

tile_get(n, x1, y1, x2, y2)

(引数)

int n, x1, y1, x2, y2

(機能)

タイル番号nのタイルパターンをグラフィック画面の(x1, y1)－(x2, y2)の領域から取り込む。

(注意)

座標(x1, y1), (x2, y2)には，オーバーサンプリングでないふつうの座標を指定すること。

取り込めるパターンの大きさには制限がある（anti.hで定義されているT_SIZEの値)。

(x1, y1)が始点で(x2, y2)が終点。大小関係を変えれば，反転したパターンも取り込める。

tone_get(n, x1, y1, x2, y2)

(引数)

int n, x1, y1, x2, y2

(機能)

トーン番号nのトーンをグラフィック画面の(x1,y1)－(x2,y2)の領域から取り込む。

(注意)

座標(x1, y1), (x2, y2)には，オーバーサンプリングでないふつうの座標を指定すること。

取り込めるパターンの大きさには制限がある（anti.hで定義されているT_SIZEの値)。

(x1, y1)が始点で(x2, y2)が終点。大小関係を変えれば，反転したパターンも取り込める。

トーンの濃さは赤成分から取る。黒に近いほど（輝度が低いほど）濃くなる。

whitepaper( )

(引数)

なし

(機能)

白紙を作る

(注意)

fill (0, 0, 511, 511, 65534) と同じ。

輝度ビットが立っていないのに注意（aa_paint ( )に支障をきたさないように)。

reverse( )

(引数)

なし

(機能)

画面を反転させる

(注意)

どうしても黒地をaa_paint ( )で塗りつぶしたいときに使う。

いったん反転させて（白地になっている）塗り，もう一度反転させて戻す。

反転に際しては輝度ビットをいじらない。

maskclear( )

(引数)

なし

(機能)

輝度ビットをすべて 0 にする。

(注意)

aa_paint ( )は，輝度ビットの立っているところでは使えない。

先にこのマスクをクリアしておくための関数。

リスト3

```
================  aa_scanconv.c  ==================
/****** アンチエリアシングつきソリッドスキャンコンバージョン ******/

#include  <graph.h>
#include  "anti.h"

/* 多角形の辺 … 線分の集合 */
typedef struct {
  int  x, y, dx2, dy2, sx, e, ry;
} EDGE;

/* リスト処理用の定数 */
#define    FIRST    0  /* アクティブエッジリストの最初の要素 */
#define    LAST     1  /* アクティブエッジリストの最後の要素*/
#define    FORWARD  0  /* アクティブエッジリストの次要素 */
#define    BACKWARD 1  /* アクティブエッジリストの前要素 */
#define    NEXT     1  /* フリーリストの次要素 */
#define    NIL     -1  /* リストの終端を示す */

#define    MAXACTIVE   128  /* 1本のスキャンラインを切れる辺の数 */
#define    MAXEDGE    1024  /* 一度に処理できる辺の数 */

EDGE  edge[ MAXEDGE ];
EDGE  *edgeptr[ MAXEDGE ], *activeedgeptr[ MAXACTIVE ];
int  activelist[ MAXACTIVE ][2]; /* アクティブエッジリストは双方向リスト・フリーリストと共用 */
int  scanlinebuffer[ MAXACTIVE ];/* 輪郭とスキャンラインの交点の x 座標 */

/* タイルおよびトーン（ペイントと共有）*/
extern unsigned char Color[3];                              /* R,G,Bの 3色 */
extern unsigned char Tile[ N_TILE ][ T_SIZE ][ T_SIZE ][3];/* R,G,Bの 3色 */
extern unsigned int  Tile_x[ N_TILE ], Tile_y[ N_TILE ];   /* タイルパターンの大きさ */
extern unsigned char Tone[ N_TONE ][ T_SIZE ][ T_SIZE ];   /* 単色 */
extern unsigned int  Tone_x[ N_TILE ], Tone_y[ N_TILE ];   /* トーンの大きさ */

extern unsigned int    Alpha[ N_PIXEL ];/* 1スキャンラインぶんの寄与率バッファ */
extern unsigned short  Slbuf[ N_PIXEL ];/* 1スキャンラインぶんのフレームバッファ */

extern unsigned char  OVERSAMPLE_NOTYET[];

extern int  tile_tone_check();  /* タイル・トーンの指定が正しいかどうか調べる */

unsigned char  TOO_COMPLEX[]="入力した点列が複雑すぎます";

FUNC  aa_scanconv( dummy )
DUMMY  dummy;
/* PTS *pts, int cmode, c/n_tile, tmode, n_tone */
{
  PTS    *pts;
  int    cmode, c, n_tile, tmode, n_tone;
  int    tile_x, tile_y, tone_x, tone_y;

  int    n_edge, y_min, y_max;
  int    n, i, j, d, fw, bk, x, x1, x2, y, y1, y2, tmp;
  int    active[2], inactive, free;
  EDGE    *tmpp;
  unsigned int  r, g, b, r1, g1, b1, v, vm, a, s;

  ARGSET( dummy );
  ARYSET(1);
  pts=PARYTOP(1);
  cmode=IVALUE(2);
  c=n_tile=IVALUE(3);
  tmode=IVALUE(4);
  n_tone=IVALUE(5);
```

```
  /* 点列のタイプ、つまり pts[0][1] は無視される ( CYCLIC として扱われる) */
  n=pts[0][0];
  if ( n>MAXEDGE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _TOO_COMPLEX,a1" );
#else
#asm
  lea.l  _TOO_COMPLEX,a1
#endasm
#endif
    return ( 1 );
  }
  if ( pts[0][2]!=OVERSAMPLE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _OVERSAMPLE_NOTYET,a1" );
#else
#asm
  lea.l  _OVERSAMPLE_NOTYET,a1
#endasm
#endif
    return ( 1 );
  }

  if ( cmode==COLOR ) {
    b1=BLUE( c );
    r1=RED( c );
    g1=GREEN( c );
  }
  i=tile_tone_check( cmode, n_tile, &tile_x, &tile_y, tmode, n_tone, &tone_x, &tone_y );
  if ( i==1 ) return ( 1 );

  y_min=N_PIXEL*OVERSAMPLE*2; y_max=-N_PIXEL*OVERSAMPLE*2;
  for ( i=1,n_edge=0; i<=n; i++ ) {
    x1 = pts[i][0];
    y1 = pts[i][1];
    if ( i!=n ) {              /* 通常の点 */
      x2 = pts[i+1][0];
      y2 = pts[i+1][1];
    } else {                   /* 終点は始点とつなげる */
      x2 = pts[1][0];
      y2 = pts[1][1];
    }
    if ( y2==y1 ) continue;  /*水平なエッジは使わない*/
    if ( y2<y1 ) {           /*スキャンコンバージョンは上から下へ処理する*/
      tmp=x1; x1=x2; x2=tmp;
      tmp=y1; y1=y2; y2=tmp;
    }
    if ( y_min>y1 ) y_min=y1;/* 図形の存在範囲を求める */
    if ( y_max<y2 ) y_max=y2;
    edge[n_edge].x = x1;     /* Bresenham アルゴリズムの初期値 */
    edge[n_edge].y = y1;
    edge[n_edge].dx2= 2*ABS( x2-x1 );
    edge[n_edge].sx= SGN( x2-x1 );
    edge[n_edge].dy2= 2*( y2-y1 );
    edge[n_edge].ry= ( y2-y1 );
    edge[n_edge].e = -( y2-y1 );
    edgeptr[n_edge]=(&edge[n_edge]);
    n_edge++;
  }
  for ( d=n_edge; d>1; ) {/* エッジを始点が上にあるものから順にソートする */
    if ( d<5 ) {           /* Shell ソートを用いている */
      d=1;
    } else {
      d=(5*d-1)/11;
    }
    for ( i=n_edge-1-d; i>=0; i-- ) {
      tmpp=edgeptr[i];    /* ポインタだけを並べ換えてスピードアップを図る */
      for ( j=i+d; j<=(n_edge-1) && ((tmpp->y)>(edgeptr[j]->y)); j+=d ) {
        edgeptr[j-d] = edgeptr[j];
      }
      edgeptr[j-d] = tmpp;
    }
  }
  free=0;                              /* フリーリストの先頭の要素 */
  active[FIRST]=active[LAST]=NIL;   /* アクティブエッジリストを空にする */
  inactive=0;                          /* アクティブでないエッジの先頭 */
  for ( i=0; i<MAXACTIVE-1; i++ ) { /* フリーリストを初期化する */
    activelist[i][NEXT]=i+1;
  }
  activelist[MAXACTIVE-1][NEXT]=NIL;/* フリーリストの終端 */

  for ( y=y_min; y<=y_max; y++ ) {
    /* 最初、またはスキャンラインの始めごとに、フレームバッファから取り込む */
    if ( y==y_min || (y%OVERSAMPLE)==0 ) {
      get( 0, PIX(y), N_PIXEL-1, PIX(y), Slbuf, N_PIXEL*sizeof(short) );
      for ( x=0; x<N_PIXEL; x++ ) Alpha[x]=0;
    }
    /* 新しい active edge を探す (始点が現在のスキャンラインと重なるエッジ) */
    while ( inactive<n_edge ) {
      if ( edgeptr[inactive]->y != y ) break;/* ソートずみなので打ち切り条件は楽 */
      if ( free==NIL ) {                         /* アクティブエッジリストの容量を超えた */
#ifdef  __GNUC__
        asm ( "  lea.l  _TOO_COMPLEX,a1" );
#else
#asm
  lea.l  _TOO_COMPLEX,a1
#endasm
#endif
        return ( 1 );
      }                             /*新しいエッジをフリーリストから取ってくる*/
      if ( active[FIRST]==NIL ) {/*アクティブエッジリストが空の場合、新規に作る*/
        active[FIRST]=active[LAST]=free;
        free=activelist[free][NEXT];
        activelist[active[LAST]][FORWARD]=activelist[active[FIRST]][BACKWARD]=NIL;
      } else {                      /*アクティブエッジリストに新しいエッジを追加する*/
        fw=free;
        free=activelist[free][NEXT];
        activelist[active[LAST]][FORWARD]=fw;
        activelist[fw][FORWARD]=NIL;
        activelist[fw][BACKWARD]=active[LAST];
        active[LAST]=fw;
      }
      activeedgeptr[active[LAST]]=edgeptr[inactive];
      inactive++;
    }
    /* アクティブエッジとスキャンラインとの交点を求める */
    n=0;
    for ( i=active[FIRST]; i!=NIL; i=activelist[i][FORWARD] ) {
      tmpp=activeedgeptr[i];
      if ( (--(tmpp->ry))<0 ) {   /* 寿命の切れたエッジはリストから削除する */
        fw=activelist[i][FORWARD];/* 双方向リストは削除が楽 */
        bk=activelist[i][BACKWARD];
        activelist[i][NEXT]=free; /* 削除したエッジはフリーリストにつなぐ */
        free=i;
        if ( fw==NIL && bk==NIL ) {
          active[FIRST]=active[LAST]=NIL;
          continue;
        }
        if ( fw==NIL ) {
          active[LAST]=bk;
          activelist[bk][FORWARD]=NIL;
          continue;
        }
        if ( bk==NIL ) {
          active[FIRST]=fw;
          activelist[fw][BACKWARD]=NIL;
          continue;
        }
        activelist[bk][FORWARD]=fw;
        activelist[fw][BACKWARD]=bk;
        continue;
      }
      scanlinebuffer[n++] = tmpp->x;/* 交点をすべてバッファに入れる */
      (tmpp->e) += (tmpp->dx2);      /* Bresenham アルゴリズムで交点を求める */
      while ( (tmpp->e) >= 0 ) {
        (tmpp->x) += (tmpp->sx);
        (tmpp->e) -= (tmpp->dy2);
      }
    }
    if ( n==0 ) continue;             /* 交点がない */
    scanlinebuffer[n]=N_PIXEL*OVERSAMPLE*2;/* 番兵 (sentinel) */
    for ( i=n-2; i>=0; i-- ) {       /* 交点をソート (番兵つき線形挿入ソート) */
      tmp=scanlinebuffer[i];
      for ( j=i+1; tmp>scanlinebuffer[j]; j++ ) {
        scanlinebuffer[j-1]=scanlinebuffer[j];
      }
      scanlinebuffer[j-1]=tmp;
    }
    if ( y<0 ) continue;              /* y でクリッピングする */
    if ( y>=(N_PIXEL*OVERSAMPLE) ) break;
                                      /* 寄与率αを更新する */
    for ( i=0; i<n-1; i+=2 ) {       /* 交点は2つひと組 */
      x1=scanlinebuffer[i]; x2=scanlinebuffer[i+1];
      if ( x2<0 ) continue;           /* x でクリッピングする */
      if ( x1>=(N_PIXEL*OVERSAMPLE) ) break;
      if ( x1<0 ) x1=0;
      if ( x2>=(N_PIXEL*OVERSAMPLE) ) x2=(N_PIXEL*OVERSAMPLE-1);
      if ( PIX(x1) == PIX(x2) ) {   /* 2つの交点が同じピクセル内にある */
        Alpha[PIX(x1)] += (x2-x1);
      } else {                        /* 2つの交点が違うピクセルにある */
        Alpha[PIX(x1)] += (OVERSAMPLE-SUBPIX(x1));
        for ( x=PIX(x1)+1; x<PIX(x2); x++ ) {
          Alpha[x] += OVERSAMPLE;
        }
        Alpha[PIX(x2)] += (SUBPIX(x2)+1);
      }
    }
    /* スキャンラインの終わりごとに、または図形の最後に、α合成して出力 */
    if ( y==(y_max-1) || SUBPIX(y)==(OVERSAMPLE-1) ) {
      vm=OVER2;                      /* αの最大値 */
      if ( tmode&ON ) {
        vm*=IMAX;                    /* トーンあり */
      }
      if ( tmode&TP ) {
        vm*=IMAX;                    /* 下地が透けるモード */
      }
      y1=PIX(y);
      for ( x1=0; x1<N_PIXEL; x1++ ) {
        a=Alpha[x1];
        if ( a==0 ) continue;
        if ( a>OVER2 ) a=OVER2;
        if ( cmode==TILE ) {       /* タイルパターン */
          b1=Tile[n_tile][y1%tile_y][x1%tile_x][0];
          r1=Tile[n_tile][y1%tile_y][x1%tile_x][1];
          g1=Tile[n_tile][y1%tile_y][x1%tile_x][2];
        }
        s=Slbuf[x1];
        v=a;
        if ( tmode&ON ) {          /* トーンあり */
          v*=Tone[n_tone][y1%tone_y][x1%tone_x];
        }
        if ( tmode&TP ) {          /* 下地が透けるモード */
          v*=VALUE(s);
        }
        b=( BLUE(s) *(vm-v)+b1*v )/vm;
        r=( RED(s)  *(vm-v)+r1*v )/vm;
        g=( GREEN(s)*(vm-v)+g1*v )/vm;
        Slbuf[x1]=RGB( r, g, b );
      }
      put( 0, y1, N_PIXEL-1, y1, Slbuf, N_PIXEL*sizeof(short) );
    }
  }
  return ( 0 );
}

FUNC   scanconv( dummy )  /* アンチエリアシングなしのバージョン */
DUMMY  dummy;             /* 速いのでちょっとした確認には使える */
/* PTS *pts, int c */
{
  PTS  *pts;
  int  c;

  int  n_edge, y_min, y_max;
  int  n, i, j, d, f, b, x1, x2, y, y1, y2, tmp;
  int  active[2], inactive, free;
  EDGE  *tmpp;

  ARGSET( dummy );
  ARYSET(1);
  pts=PARYTOP(1);
  c=IVALUE(2);

  n=pts[0][0];
  if ( n>MAXEDGE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _TOO_COMPLEX,a1" );
#else
#asm
  lea.l  _TOO_COMPLEX,a1
#endasm
#endif
    return ( 1 );
  }
  if ( pts[0][2]!=OVERSAMPLE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _OVERSAMPLE_NOTYET,a1" );
#else
#asm
  lea.l  _OVERSAMPLE_NOTYET,a1
#endasm
#endif
    return ( 1 );
  }
  y_min=N_PIXEL*2; y_max=-N_PIXEL*2;
  for ( i=1,n_edge=0; i<=n; i++ ) {
    x1 = PIX(pts[i][0]);
    y1 = PIX(pts[i][1]);
    if ( i!=n ) {
      x2 = PIX(pts[i+1][0]);
```

```
      y2 = PIX(pts[i+1][1]);
    } else {
      x2 = PIX(pts[1][0]);
      y2 = PIX(pts[1][1]);
    }
    if ( y2==y1 ) continue;
    if ( y2<y1 ) {
      tmp=x1; x1=x2; x2=tmp;
      tmp=y1; y1=y2; y2=tmp;
    }
    if ( y_min>y1 ) y_min=y1;
    if ( y_max<y2 ) y_max=y2;
    edge[n_edge].x = x1;
    edge[n_edge].y = y1;
    edge[n_edge].dx2= 2*ABS( x2-x1 );
    edge[n_edge].sx= SGN( x2-x1 );
    edge[n_edge].dy2= 2*( y2-y1 );
    edge[n_edge].ry= ( y2-y1 );
    edge[n_edge].e = -( y2-y1 );
    edgeptr[n_edge]=(&edge[n_edge]);
    n_edge++;
  }
  for ( d=n_edge; d>1; ) {
    if ( d<5 ) {
      d=1;
    } else {
      d=(5*d-1)/11;
    }
    for ( i=n_edge-1-d; i>=0; i-- ) {
      tmpp=edgeptr[i];
      for ( j=i+d; j<=(n_edge-1) && ((tmpp->y)>(edgeptr[j]->y)); j+=d ) {
        edgeptr[j-d] = edgeptr[j];
      }
      edgeptr[j-d] = tmpp;
    }
  }
  free=0;
  active[FIRST]=active[LAST]=NIL;
  inactive=0;
  for ( i=0; i<MAXACTIVE-1; i++ ) {
    activelist[i][NEXT]=i+1;
  }
  activelist[MAXACTIVE-1][NEXT]=NIL;

  for ( y=y_min; y<=y_max; y++ ) {
    while ( inactive<n_edge ) {
      if ( edgeptr[inactive]->y != y ) break;
      if ( free==NIL ) {
#ifdef  __GNUC__
        asm ( "  lea.l  _TOO_COMPLEX,a1" );
#else
#asm
  lea.l  _TOO_COMPLEX,a1
#endasm
#endif
        return ( 1 );
      }
      if ( active[FIRST]==NIL ) {
        active[FIRST]=active[LAST]=free;
        free=activelist[free][NEXT];
        activelist[active[LAST]][FORWARD]=activelist[active[FIRST]][BACKWARD]=NIL;
      } else {
        f=free;
        free=activelist[free][NEXT];
        activelist[active[LAST]][FORWARD]=f;
        activelist[f][FORWARD]=NIL;
        activelist[f][BACKWARD]=active[LAST];
        active[LAST]=f;
      }
      activeedgeptr[active[LAST]]=edgeptr[inactive];
      inactive++;
    }
    n=0;
    for ( i=active[FIRST]; i!=NIL; i=activelist[i][FORWARD] ) {
      tmpp=activeedgeptr[i];
      if ( (--(tmpp->ry))<0 ) {
        f=activelist[i][FORWARD];
        b=activelist[i][BACKWARD];
        activelist[i][NEXT]=free;
        free=i;
        if ( f==NIL && b==NIL ) {
          active[FIRST]=active[LAST]=NIL;
          continue;
        }
        if ( f==NIL ) {
          active[LAST]=b;
          activelist[b][FORWARD]=NIL;
          continue;
        }
        if ( b==NIL ) {
          active[FIRST]=f;
          activelist[f][BACKWARD]=NIL;
          continue;
        }
        activelist[b][FORWARD]=f;
        activelist[f][BACKWARD]=b;
        continue;
      }
      scanlinebuffer[n++] = tmpp->x;
      (tmpp->e) += (tmpp->dx2);
      while ( (tmpp->e) >= 0 ) {
        (tmpp->x) += (tmpp->sx);
        (tmpp->e) -= (tmpp->dy2);
      }
    }
    if ( n==0 ) continue;
    scanlinebuffer[n]=N_PIXEL*2;
    for ( i=n-2; i>=0; i-- ) {
      tmp=scanlinebuffer[i];
      for ( j=i+1; tmp>scanlinebuffer[j]; j++ ) {
        scanlinebuffer[j-1]=scanlinebuffer[j];
      }
      scanlinebuffer[j-1]=tmp;
    }
    for ( i=0; i<n-1; i+=2 ) {
      line( scanlinebuffer[i], y, scanlinebuffer[i+1], y, c, 0xFFFF );
    }
  }
  return ( 0 );
}
```

リスト4

```
================ aa_lines.c ==================
/****** アンチエリアシング付き点列描画 ******/
#include  <graph.h>
#include  <math.h>
#include  "anti.h"
extern unsigned char  Color[3];        /* 描画色 */
extern unsigned int   Alpha[ N_PIXEL ];/* 1ラインぶんの寄与率バッファ */
extern unsigned short Slbuf[ N_PIXEL ];/* 1ラインぶんのフレームバッファ */
extern unsigned char  OVERSAMPLE_NOTYET[];
int  Direction;                        /*描画中の点列の傾きが大きいか小さいか*/
void  aa_line( x1, y1, x2, y2, w )     /* 1区間だけ(線分1本)描く */
int   x1, y1, x2, y2, w;
{
  double    h0;
  int    newdirection, i, j, x, y, sx, dx, dx2, sy, dy, dy2, e, h;
  int    xa, xb, ya, yb;
  unsigned int  r, g, b;
  unsigned int  a;
  unsigned short  c, s;
  x=x1;                                /*Bresenhamアルゴリズムの初期設定*/
  y=y1;
  dx=ABS( x2-x1 );
  sx=SGN( x2-x1 );
  dx2=dx*2;
  dy=ABS( y2-y1 );
  sy=SGN( y2-y1 );
  dy2=dy*2;
  if ( dx==0 && dy==0 ) return;
  e=0;
  c=RGB( Color[1], Color[2], Color[0] );
  if ( dx>dy ) {                       /*傾きが小さい場合は x でループ*/
    if ( sx>0 ) {
      newdirection=1;
    } else {
      newdirection=3;
    }
  } else {                             /*傾きが大きい場合は y でループ*/
    if ( sy>0 ) {
      newdirection=2;
    } else {
      newdirection=4;
    }
  }
  if ( newdirection==1 || newdirection==3 ) {/*傾きが小さい場合は x でループ*/
    /* Direcrion が切り換わるとき、スクリーンから1ライン取り込む */
    if ( Direction != newdirection ) {        /*Direction=0(最初)の場合も含む*/
      get( PIX(x), 0, PIX(x), N_PIXEL-1, Slbuf, N_PIXEL*sizeof(short) );
      for ( i=0; i<N_PIXEL; i++ ) Alpha[i]=0;/*寄与率バッファをクリアする */
      Direction=newdirection;
    }
    /* 幅から高さを計算する */
    h0 = ( (double)w * sqrt((double)dx*(double)dx+(double)dy*(double)dy )/dx);
    h = (int)h0;
    y -= (int)(h0/2.0);                /* 下端線分の始点 */
    for ( i=0; i<dx; i++, x+=sx ) {    /* メインループ */
      /* 1ライン処理するごとにスクリーンから新しいラインを取り込む */
      /* 始点ではラインを取り込むとは限らない(前の線分の情報を残す)*/
      if ( (sx==1 && SUBPIX(x)==0) || (sx==-1 && SUBPIX(x)==(OVERSAMPLE-1))) {
        get( PIX(x), 0, PIX(x), N_PIXEL-1, Slbuf, N_PIXEL*sizeof(short) );
        for ( j=0; j<N_PIXEL; j++ ) Alpha[j]=0;/* 寄与率バッファをクリアする */
      }
      ya=y; yb=y+h-1;                  /* 寄与率αを更新する */
      if ( yb>=0 && ya<(N_PIXEL*OVERSAMPLE) ) {
        if ( ya<0 ) ya=0;
        if ( yb>=(N_PIXEL*OVERSAMPLE) ) yb=(N_PIXEL*OVERSAMPLE-1);
        if ( PIX(ya) == PIX(yb) ) {
          Alpha[PIX(ya)] += (yb-ya);
        } else {
          Alpha[PIX(ya)] += (OVERSAMPLE-SUBPIX(ya));
          for ( j=(PIX(ya)+1); j<PIX(yb); j++ ) {
            Alpha[j] += OVERSAMPLE;
          }
          Alpha[PIX(yb)] += (SUBPIX(yb)+1);
        }
      }
      e+=dy2;    /* Bresenham アルゴリズムで下端線分を発生する */
      if ( e>dx ) {
        y+=sy;
        e-=dx2;
      }
      /* 1ラインごと(または終点)でα合成と出力を行う */
      if ( i==(dx-1) || (sx==-1 && SUBPIX(x)==0)||(sx==1 && SUBPIX(x)==(OVERSAMPLE-1))) {
        for ( j=0; j<N_PIXEL; j++ ) {
          if ( (a=Alpha[j])==0 ) continue;
          if ( a>=OVER2 ) {  /* 寄与率1ならベタ書き */
            Slbuf[j] = c;
          } else {           /* そうでないならα合成 */
            s=Slbuf[j];
            b=( (OVER2-a)*BLUE(s) + a*Color[0]) /OVER2;
            r=( (OVER2-a)*RED(s)  + a*Color[1]) /OVER2;
            g=( (OVER2-a)*GREEN(s)+ a*Color[2]) /OVER2;
            Slbuf[j] = RGB( r, g, b );
          }
        }
        put( PIX(x), 0, PIX(x), N_PIXEL-1, Slbuf, N_PIXEL*sizeof(short) );
      }
    }
  } else {  /* 傾きが大きい場合は y でループ、以下同様 */
    if ( Direction != newdirection ) {
      get( 0, PIX(y), N_PIXEL-1, PIX(y), Slbuf, N_PIXEL*sizeof(short) );
      for ( i=0; i<N_PIXEL; i++ ) Alpha[i]=0;
      Direction=newdirection;
    }
    h0 = ( (double)w * sqrt( (double)dx*(double)dx+(double)dy*(double)dy ) / dy );
    h = (int)h0;
    x -= (int)(h0/2.0);
    for ( i=0; i<dy; i++, y+=sy ) {
      if ( (sy==1 && SUBPIX(y)==0) || (sy==-1 && SUBPIX(y)==(OVERSAMPLE-1))) {
        get( 0, PIX(y), N_PIXEL-1, PIX(y), Slbuf, N_PIXEL*sizeof(short) );
        for ( j=0; j<N_PIXEL; j++ ) Alpha[j]=0;
      }
      xa=x; xb=x+h-1;
      if ( xb>=0 && xa<(N_PIXEL*OVERSAMPLE) ) {
        if ( xa<0 ) xa=0;
        if ( xb>=(N_PIXEL*OVERSAMPLE) ) xb=(N_PIXEL*OVERSAMPLE-1);
        if ( PIX(xa) == PIX(xb) ) {
          Alpha[PIX(xa)] += (xb-xa);
        } else {
```


▶鳥大の電算研に入って,いまレイトレやってます。うーん,時間がかかるなあ(PC-9801+コプロ+フレームバッファ)。てことはX68000(2MRAM, コプロなし)だったら死ぬほどおそいんだろうなあ。うーん。 森下 寛和(18)鳥取県

```
          Alpha[PIX(xa)] += (OVERSAMPLE-SUBPIX(xa));
          for ( j=(PIX(xa)+1); j<PIX(xb); j++ ) {
            Alpha[j] += OVERSAMPLE;
          }
          Alpha[PIX(xb)] += (SUBPIX(xb)+1);
        }
      }
      e+=dx2;
      if ( e>dy ) {
        x+=sx;
        e-=dy2;
      }
      if ( i==(dy-1) || (sy==-1 && SUBPIX(y)==0) || (sy==1 && SUBPIX(y)==(OVERSAMPLE-1))) {
        for ( j=0; j<N_PIXEL; j++ ) {
          if ( (a=Alpha[j])==0 ) continue;
          if ( a>=OVER2 ) {
            Slbuf[j]=c;
          } else {
            s=Slbuf[j];
            b=( (OVER2-a)*BLUE(s) + a*Color[0]) /OVER2;
            r=( (OVER2-a)*RED(s)  + a*Color[1]) /OVER2;
            g=( (OVER2-a)*GREEN(s)+ a*Color[2]) /OVER2;
            Slbuf[j] = RGB( r, g, b );
          }
        }
        put( 0, PIX(y), N_PIXEL-1, PIX(y), Slbuf, N_PIXEL*sizeof(short) );
      }
    }
  }
  return;
}
FUNC  aa_lines( dummy )  /* 関数本体、全点列を描画する */
DUMMY  dummy;
/* PTS *pts, int c */
{
  PTS  *pts;
  int  c;
  int  i, n;
  ARGSET( dummy );
  ARYSET(1);
  pts=PARYTOP(1);
  c=IVALUE(2);
  Color[0]=BLUE(c);
  Color[1]=RED(c);
  Color[2]=GREEN(c);
  n=pts[0][0];
  if ( pts[0][2]!=OVERSAMPLE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _OVERSAMPLE_NOTYET,a1" );
#else
#asm
  lea.l  _OVERSAMPLE_NOTYET,a1
#endasm
#endif
    return ( 1 );
  }
  Direction=0;             /* 始めは傾きの大小が不明 */
  for ( i=1; i<n; i++ ) {
    aa_line( pts[i][0], pts[i][1], pts[i+1][0], pts[i+1][1], pts[i][2] );
  }
  if ( pts[0][1]==CYCLIC ) {  /* 点列が循環している場合、終点と始点をつなぐ */
    aa_line( pts[n][0], pts[n][1], pts[1][0], pts[1][1], pts[n][2] );
  }
  return ( 0 );
}
FUNC  lines( dummy )    /* アンチエリアシングなしのバージョン */
DUMMY  dummy;           /* 速いのでちょっとした確認には使える */
/* PTS *pts, int c */
{
  PTS  *pts;
  int  c;
  int  i, n;
  ARGSET( dummy );
  ARYSET(1);
  pts=PARYTOP(1);
  c=IVALUE(2);
  n=pts[0][0];
  if ( pts[0][2]!=OVERSAMPLE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _OVERSAMPLE_NOTYET,a1" );
#else
#asm
  lea.l  _OVERSAMPLE_NOTYET,a1
#endasm
#endif
    return ( 1 );
  }
  for ( i=1; i<n; i++ ) {
    line( PIX(pts[i][0]), PIX(pts[i][1]), PIX(pts[i+1][0]), PIX(pts[i+1][1]), c, 0xFFFF );
  }
  if ( pts[0][1]==CYCLIC ) {
    line( PIX(pts[n][0]), PIX(pts[n][1]), PIX(pts[1][0]), PIX(pts[1][1]), c, 0xFFFF );
  }
  return ( 0 );
}
```

リスト5

```
================  aa_paint.c  ==================
/****** アンチエリアシング対応ペイントルーチン ******/
#include  <graph.h>
#include  "anti.h"
/* キュー(待ち行列、または FIFO バッファ) */
#define  MAXQUEUE  1024
short    Qx[ MAXQUEUE ], Qy[ MAXQUEUE ];
int      QPi, QPo;
#define  INIT_QUEUE  { QPi=QPo=0; }
#define  ENQUEUE( X, Y ) { Qx[ QPi ]=(X); Qy[ QPi ]=(Y); QPi=(++QPi)%MAXQUEUE; }
#define  DEQUEUE( X, Y ) { (X)=Qx[ QPo ]; (Y)=Qy[ QPo ]; QPo=(++QPo)%MAXQUEUE; }
#define  EMPTY_QUEUE  (QPi==QPo)
/* タイルおよびトーン(ソリッドスキャンコンバージョンと共有) */
extern unsigned char Color[3];                                /*R,G,B の 3色*/
extern unsigned char Tile[ N_TILE ][ T_SIZE ][ T_SIZE ][3];/*R,G,B の 3色*/
extern unsigned int  Tile_x[ N_TILE ], Tile_y[ N_TILE ];   /*タイルパターンの大きさ*/
extern unsigned char Tone[ N_TONE ][ T_SIZE ][ T_SIZE ];   /*単色 */
extern unsigned int  Tone_x[ N_TILE ], Tone_y[ N_TILE ];   /*トーンの大きさ*/
extern int  tile_tone_check();       /* タイル・トーンの指定が正しいかどうか調べる */
extern unsigned short  Slbuf[ N_PIXEL ];          /*1スキャンラインぶんのフレームバッファ*/
unsigned short  slubuf[512], slmbuf[512], sldbuf[512]; /*ペイント用スキャンラインバッファ*/
unsigned char  OUTOF_SCREEN[]="指定した座標がスクリーンの範囲外です";
FUNC  aa_paint( dummy )                                        /* 関数本体 */
DUMMY  dummy;
/* int  x0, y0, c */
{
  int    x0, y0, cmode, c, n_tile, tmode, n_tone;
  int    tile_x, tile_y, tone_x, tone_y;
  int    i, x, y, x1, x2;
  int    sign, sign1;
  unsigned int  r, g, b, r1, g1, b1, v, vm, s;
  ARGSET( dummy );
  ARYSET(1);
  x0=IVALUE(1);
  y0=IVALUE(2);
  cmode=IVALUE(3);
  c=n_tile=IVALUE(4);
  tmode=IVALUE(5);
  n_tone=IVALUE(6);
  if ( x0<0 || x0>=N_PIXEL || y0<0 || y0>=N_PIXEL ) {  /* 画面外は塗れない */
#ifdef  __GNUC__
    asm ( "  lea.l  _OUTOF_SCREEN,a1" );
#else
#asm
  lea.l  _OUTOF_SCREEN,a1
#endasm
#endif
    return ( 1 );
  }
  if ( cmode==COLOR ) {  /* タイルパターンを使わないなら描画色は一定 */
    b1=BLUE( c );
    r1=RED( c );
    g1=GREEN( c );
  }
  i=tile_tone_check( cmode, n_tile, &tile_x, &tile_y, tmode, n_tone, &tone_x, &tone_y);
  if ( i==1 ) return ( 1 );
  INIT_QUEUE;             /* FIFO バッファを空にする */
  ENQUEUE( x0, y0 );     /* シードを FIFO バッファに入れる */
  vm=VALUE( point( x0, y0 ) );  /* 寄与率の最大値は出発点を100%とする */
  if ( tmode&ON ) {      /* トーンを考慮した寄与率の最大値 */
    vm*=IMAX;
  }
  while ( !EMPTY_QUEUE ) {  /* ペイント終了条件 */
    DEQUEUE( x0, y0 );       /* シードを FIFO バッファから取り出す */
    if ( point( x0, y0 )&FMASK ) continue;          /*ペイントずみ */
    if ( point( x0, y0 )&VMASK == 0 ) continue; /*黒いところには塗っても仕方がない*/
    if ( y0>0 )  /* 上への到達可能性を調べるためのバッファ */
      get( 0, y0-1, N_PIXEL-1, y0-1, slubuf, N_PIXEL*sizeof(short) );
    get( 0, y0  , N_PIXEL-1, y0  , Slbuf , N_PIXEL*sizeof(short) );
    if ( y0<N_PIXEL-1 )  /* 下への到達可能性を調べるためのバッファ */
      get( 0, y0+1, N_PIXEL-1, y0+1, sldbuf, N_PIXEL*sizeof(short) );
    for ( x=0; x<N_PIXEL; x++ ) {
      slubuf[x] &= SLMASK;
      slmbuf[x] = Slbuf[x]&SLMASK;            /*左右への到達可能性を調べるバッファ*/
      sldbuf[x] &= SLMASK;
      if ( slubuf[x]==0 ) slubuf[x]=FMASK;  /*黒いところには塗っても仕方ないので*/
      if ( slmbuf[x]==0 ) slubuf[x]=FMASK;  /*始めからペイントずみにしておく */
      if ( sldbuf[x]==0 ) slubuf[x]=FMASK;
    }
    for ( x1=x0; x1>=0; x1-- ) {  /* 左へ塗り進める */
      if ( cmode==TILE ) {        /* タイルパターン */
        b1=Tile[n_tile][y0%tile_y][x1%tile_x][0];
        r1=Tile[n_tile][y0%tile_y][x1%tile_x][1];
        g1=Tile[n_tile][y0%tile_y][x1%tile_x][2];
      }
      if ( tmode&ON ) {           /* トーンあり */
        v=(slmbuf[x1]>>VSHIFT)*Tone[n_tone][y0%tone_y][x1%tone_x];
      } else {                    /* トーンなし */
        v=slmbuf[x1]>>VSHIFT;
      }
      if ( tmode&TP ) {           /* 下地が透けて見える */
        s=Slbuf[x1];
        b=( BLUE(s)*(vm-v) + b1*v )/vm;
        r=(  RED(s)*(vm-v) + r1*v )/vm;
        g=(GREEN(s)*(vm-v) + g1*v )/vm;
      } else {    /* 下地は無視 */
        b=b1*v/vm;
        r=r1*v/vm;
        g=g1*v/vm;
      }
      Slbuf[x1]=RGBI( r, g, b, FMASK );/* ペイントずみフラグを立てる */
      if ( x1==0 ) break;                /* さらに左に進めるか調べる */
      if ( slmbuf[x1-1]&FMASK ) break; /* ペイントずみのところで止める */
      if ( slmbuf[x1]<slmbuf[x1-1] ) break;  /* 次が明るくなりそうなら止める */
    }
    for ( x2=x0; x2<N_PIXEL; x2++ ) {  /* 右へ塗り進める、以下同様 */
      if ( cmode==TILE ) {
        b1=Tile[n_tile][y0%tile_y][x2%tile_x][0];
        r1=Tile[n_tile][y0%tile_y][x2%tile_x][1];
        g1=Tile[n_tile][y0%tile_y][x2%tile_x][2];
      }
      if ( tmode&ON ) {
        v=(slmbuf[x2]>>VSHIFT)*Tone[n_tone][y0%tone_y][x2%tone_x];
      } else {
        v=slmbuf[x2]>>VSHIFT;
      }
      if ( tmode&TP ) {
        s=Slbuf[x2];
        b=( BLUE(s)*(vm-v) + b1*v )/vm;
        r=(  RED(s)*(vm-v) + r1*v )/vm;
        g=(GREEN(s)*(vm-v) + g1*v )/vm;
      } else {
        b=b1*v/vm;
        r=r1*v/vm;
        g=g1*v/vm;
      }
      Slbuf[x2]=RGBI( r, g, b, FMASK );
      if ( x2==N_PIXEL-1 ) break;
      if ( slmbuf[x2+1]&FMASK ) break;
      if ( slmbuf[x2]<slmbuf[x2+1] ) break;
    }
    if ( y0>0 ) {  /* 上へ塗り進める可能性を調べる */
    /*sign は輝度の勾配 */
    /*ひとつ上のスキャンラインの輝度が極大になる (sign が + から - に転じる) ところで塗り進める */
    /*塗り進めない領域から途中で出た場合には、その直前で塗り進める */
      sign=1;
      for ( x=x1; x<=x2; x++ ) {
        if ( slubuf[x]&FMASK || slmbuf[x]<slubuf[x] ) {
          sign=1;
          continue;
        }
        if ( x==x2 || slubuf[x+1]&FMASK || slmbuf[x+1]<slubuf[x+1] ) {
```

▶そうか,だからPIC.FNCを組み込もうとしたらおかしくなったんだな。ディスクの誤植かと思った。
坂井　純一(20)茨城県

```
          if ( sign>0 ) ENQUEUE( x, y0-1 );
          continue;
        }
        sign1=slubuf[x+1]-slubuf[x];
        if ( sign>0 && sign1<0 ) ENQUEUE( x, y0-1 );
        if ( sign*sign1<0 ) sign=sign1;
      }
    }
    if ( y0<N_PIXEL-1 ) {  /* 下へ塗り進める可能性を調べる、以下同様 */
      sign=1;
      for ( x=x1; x<=x2; x++ ) {
        if ( sldbuf[x]&FMASK || slmbuf[x]<sldbuf[x] ) {
          sign=1;
          continue;
        }
        if ( x==x2 || sldbuf[x+1]&FMASK || slmbuf[x+1]<sldbuf[x+1] ) {
          if ( sign>0 ) ENQUEUE( x, y0+1 );
          continue;
        }
        sign1=sldbuf[x+1]-sldbuf[x];
        if ( sign>0 && sign1<0 ) ENQUEUE( x, y0+1 );
        if ( sign*sign1<0 ) sign=sign1;
      }
    }
    put( 0, y0, N_PIXEL-1, y0, Slbuf, N_PIXEL*sizeof(short) );
  }
  return ( 0 );
}
```

リスト6

```
================ aa_procs.c ==================
/****** アンチエリアシング関係の処理（タイル・トーンなど）******/
#include  "anti.h"
/* 色（カラーコードまたはタイルパターン）*/
unsigned char Color[3] ;                            /*R,G,B の 3色 */
unsigned char Tile[ N_TILE ][ T_SIZE ][ T_SIZE ][3];/*R,G,B の 3色 */
unsigned int  Tile_x[ N_TILE ], Tile_y[ N_TILE ];   /*タイルパターンの大きさ*/

/* トーン */
unsigned char  Tone[ N_TONE ][ T_SIZE ][ T_SIZE ];  /* 単色 */
unsigned int   Tone_x[ N_TONE ], Tone_y[ N_TONE ];  /* トーンの大きさ */
unsigned int   Alpha[ N_PIXEL ];      /*1スキャンラインぶんの寄与率バッファ*/
unsigned short Slbuf[ N_PIXEL ];      /*1スキャンラインぶんのフレームバッファ*/
unsigned short Temp[ T_SIZE*T_SIZE ];/* tile_get(), tone_get() 用のテンポラリ配列*/
unsigned char  TOOMANY_PATTERN[]="パターンの番号が大きすぎます";
unsigned char  OVERSIZE[]="パターンのサイズが大きすぎます";

FUNC  tile_get( dummy )
DUMMY  dummy;
/* int n, int x1, int y1, int x2, int y2 */
{
  int    n, x1, y1, x2, y2;
  int    i, j, dx, dy, sx, sy, x, y;
  unsigned short  c;
  ARGSET( dummy );
  n=IVALUE(1);
  x1=IVALUE(2);
  y1=IVALUE(3);
  x2=IVALUE(4);
  y2=IVALUE(5);
  if ( n>=N_TILE ) {
#ifdef  __GNUC__
     asm( "  lea.l  _TOOMANY_PATTERN,a1" );
#else
#asm
  lea.l  _TOOMANY_PATTERN,a1
#endasm
#endif
     return ( 1 );
  }
  dx = ABS( x2-x1 )+1;
  dy = ABS( y2-y1 )+1;
  sx = SGN( x2-x1 );
  sy = SGN( y2-y1 );
  if ( dx>T_SIZE || dy>T_SIZE ) {
#ifdef  __GNUC__
     asm ( "  lea.l  _OVERSIZE,a1" );
#else
#asm
     lea.l  _OVERSIZE,a1
#endasm
#endif
     return ( 1 );
  }
  Tile_x[n] = dx;
  Tile_y[n] = dy;
  get( MIN(x1,x2), MIN(y1,y2), MAX(x1,x2), MAX(y1,y2), Temp, dx*dy*sizeof(short) );
  for ( i=0, y=((sy>0)?(0):(dy-1)); i<dy; i++, y+=sy ) {
    for ( j=0, x=((sx>0)?(0):(dx-1)); j<dx; j++, x+=sx ) {
      c=Temp[y*dx+x];
      Tile[n][i][j][0]=BLUE(c);
      Tile[n][i][j][1]=RED(c);
      Tile[n][i][j][2]=GREEN(c);
    }
  }
  return ( 0 );
}

FUNC  tone_get( dummy )
DUMMY  dummy;
/* int n, int x1, int y1, int x2, int y2 */
{
  int    n, x1, y1, x2, y2;
  int    i, j, dx, dy, sx, sy, x, y;
  unsigned short  c;

  ARGSET( dummy );
  n=IVALUE(1);
  n=IVALUE(1);
  x1=IVALUE(2);
  y1=IVALUE(3);
  x2=IVALUE(4);
  y2=IVALUE(5);
  if ( n>=N_TONE ) {
#ifdef  __GNUC__
     asm( "  lea.l  _TOOMANY_PATTERN,a1" );
#else
#asm
  lea.l  _TOOMANY_PATTERN,a1
#endasm
#endif
     return ( 1 );
  }

  dx = ABS( x2-x1 )+1;
  dy = ABS( y2-y1 )+1;
  sx = SGN( x2-x1 );
  sy = SGN( y2-y1 );

  if ( dx>T_SIZE || dy>T_SIZE ) {
#ifdef  __GNUC__
     asm ( "  lea.l  _OVERSIZE,a1" );
#else
#asm
  lea.l  _OVERSIZE,a1
#endasm
#endif
     return ( 1 );
  }
  Tone_x[n] = dx;
  Tone_y[n] = dy;
  get( MIN(x1,x2), MIN(y1,y2), MAX(x1,x2), MAX(y1,y2), Temp, dx*dy*sizeof(short) );
  for ( i=0, y=((sy>0)?(0):(dy-1)); i<dy; i++, y+=sy ) {
    for ( j=0, x=((sx>0)?(0):(dx-1)); j<dx; j++, x+=sx ) {
      c=Temp[y*dx+x];
      Tone[n][i][j]=(IMAX-RED(c));  /* 黒い部分ほど濃くする */
    }
  }
  return ( 0 );
}
unsigned char  ILLEGAL_NTILE[]="タイルパターン番号が不正です";
unsigned char  TILE_TOO_LARGE[]="タイルパターンのサイズが大きすぎます";
unsigned char  ILLEGAL_CMODE[]="色/タイルのモードを正しく指定して下さい";
unsigned char  ILLEGAL_NTONE[]="トーン番号が不正です";
unsigned char  TONE_TOO_LARGE[]="トーンのサイズが大きすぎます";
unsigned char  ILLEGAL_TMODE[]="トーンのモードを正しく指定して下さい";
int  tile_tone_check( cmode, n_tile, tile_x, tile_y, tmode, n_tone, tone_x, tone_y )
int  cmode, n_tile, *tile_x, *tile_y, tmode, n_tone, *tone_x, *tone_y;
{
  switch ( cmode ) {
  case COLOR:
    break;
  case TILE:
    if ( n_tile >= N_TILE ) {
#ifdef  __GNUC__
      asm ( "  lea.l  _ILLEGAL_NTILE,a1" );
#else
#asm
  lea.l  _ILLEGAL_NTILE,a1
#endasm
#endif
      return ( 1 );
    }
    *tile_x=Tile_x[n_tile];
    *tile_y=Tile_y[n_tile];
    if ( *tile_x>T_SIZE || *tile_y>T_SIZE ) {
#ifdef  __GNUC__
      asm ( "  lea.l  _TILE_TOO_LARGE,a1" );
#else
#asm
  lea.l  _TILE_TOO_LARGE,a1
#endasm
#endif
      return ( 1 );
    }
    break;
  default:
#ifdef  __GNUC__
    asm ( "  lea.l  _ILLEGAL_CMODE,a1" );
#else
#asm
  lea.l  _ILLEGAL_CMODE,a1
#endasm
#endif
    return ( 1 );
    break;
  }
  switch ( tmode ) {
  case ON_TP:
  case ON_NTP:
    if ( n_tone >= N_TONE ) {
#ifdef  __GNUC__
      asm ( "  lea.l  _ILLEGAL_NTONE,a1" );
#else
#asm
  lea.l  _ILLEGAL_NTONE,a1
#endasm
#endif
      return ( 1 );
    }
    *tone_x=Tone_x[n_tone];
    *tone_y=Tone_y[n_tone];
    if ( *tone_x>T_SIZE || *tone_y>T_SIZE ) {
#ifdef  __GNUC__
      asm ( "  lea.l  _TONE_TOO_LARGE,a1" );
#else
#asm
  lea.l  _TONE_TOO_LARGE,a1
#endasm
#endif
      return ( 1 );
    }
    break;
  case OFF_TP:
  case OFF_NTP:
    break;
  default:
#ifdef  __GNUC__
    asm ( "  lea.l  _ILLEGAL_TMODE,a1" );
#else
#asm
  lea.l  _ILLEGAL_TMODE,a1
#endasm
#endif
    return ( 1 );
    break;
  }
  return ( 0 );
}

FUNC  whitepaper()
{
  int  i;
  for ( i=0; i<N_PIXEL; i++ ) {
    Slbuf[i]=RGBI( IMAX, IMAX, IMAX, 0 );
  }
  for ( i=0; i<N_PIXEL; i++ ) {
    put( 0, i, N_PIXEL-1, i, Slbuf, N_PIXEL*sizeof(short) );
```


▶X68000本体が買えないのでモニタだけ買ってしまおうと思っている私は悲しい。
棚橋　計介(23)岐阜県

```
  }
  return ( 0 );
}

FUNC  reverse()
{
  int  i, j;
  for ( i=0; i<N_PIXEL; i++ ) {
    get( 0, i, N_PIXEL-1, i, Slbuf, N_PIXEL*sizeof(short) );
    for ( j=0; j<N_PIXEL; j++ ) {
      Slbuf[j] ^= RGBI( IMAX, IMAX, IMAX, 0 );
    }
    put( 0, i, N_PIXEL-1, i, Slbuf, N_PIXEL*sizeof(short) );
  }
  return ( 0 );
}

FUNC  maskclear()
{
  int  i, j;
  for ( i=0; i<N_PIXEL; i++ ) {
    get( 0, i, N_PIXEL-1, i, Slbuf, N_PIXEL*sizeof(short) );
    for ( j=0; j<N_PIXEL; j++ ) {
      Slbuf[j] &= RGBI( IMAX, IMAX, IMAX, 0 );
    }
    put( 0, i, N_PIXEL-1, i, Slbuf, N_PIXEL*sizeof(short) );
  }
  return ( 0 );
}
FUNC  monotone()
{
  int    i, j;
  unsigned int  s, c;
  for ( i=0; i<N_PIXEL; i++ ) {
    get( 0, i, N_PIXEL-1, i, Slbuf, N_PIXEL*sizeof(short) );
    for ( j=0; j<N_PIXEL; j++ ) {
      s=Slbuf[j];
      c=(RED(s)*77+GREEN(s)*151+BLUE(s)*28)/256;
      Slbuf[j] = RGB( c, c, c );
    }
    put( 0, i, N_PIXEL-1, i, Slbuf, N_PIXEL*sizeof(short) );
  }
  return ( 0 );
}
```

リスト7

```
================  main.c  ==================
/****** パラメータ受け渡し用仮変数の実体 ******/

unsigned short  *par;         /* 一時的な引数リスト */
unsigned short  *ary[10+1];   /* 一時的な配列リスト：X-BASIC の引数は最大 10 個 */

/****** コンパイラを通すためのダミー ( 実行されない ) ******/

void  main()
{
}
```

リスト8

```
===============  pts_curve.c  ==================
/****** Bezier曲線を使った内挿により自由曲線を発生する ******/
#include  <math.h>
#include  "anti.h"
#define MAXPTS1    256                       /*入力点列の長さの最大値*/
#define SCALE       32                       /*整数演算の精度を確保するための倍率*/
#define MIN_LENGTH (8*OVERSAMPLE*SCALE) /*再帰分割を打ち切る制御点の間隔*/
typedef int    ivector[2];                   /*整数値ベクトル */
typedef double  vector[2];                   /*実数値ベクトル */
ivector Ptstemp[MAXPTS1][3];                 /*テンポラリの制御点 */
int  Type_pts, N_pts1, N_pts2, Maxpts2;
PTS  *pts2;                                  /*出力の点列 */
#define  CPY( V1, V2 ) { V1[0]=V2[0]; V1[1]=V2[1]; }   /* ベクトルのコピー */
#define  SCPY( V1, V2) { V1[0]=V2[0]/SCALE; V1[1]=V2[1]/SCALE; } /*倍率を加味したベクトルのコピー
*/
#define  LENGTH( V )  ( sqrt( V[0]*V[0]+V[1]*V[1] ) )  /* ベクトルの長さ */
void  normalize( v1, v2 )                    /* 単位ベクトル化 */
vector  v1, v2;
{
  double  l;
  l=LENGTH( v1 );
  v2[0] = v1[0]/l;
  v2[1] = v1[1]/l;
  return;
}
void  mult_factor_vec( v1, v2 )      /*方向は変えないで長さを同じにする */
vector  v1, v2;
{
  double  factor;
  factor=LENGTH( v2 )/LENGTH( v1 );
  v2[0] = factor*v1[0];
  v2[1] = factor*v1[1];
  return;
}
void  control( p1, p2, p3 )          /*サンプル点から制御点を発生する */
ivector  p1, p2, p3;
{
  vector  va, vb, vc, norma, normb;
  double  la, lb;
  int  i, j, k;
  va[0] = (double)(p2[0]-p1[0]);    /* 隣のサンプル点への方向ベクトル */
  va[1] = (double)(p2[1]-p1[1]);
  vb[0] = (double)(p3[0]-p2[0]);
  vb[1] = (double)(p3[1]-p2[1]);
  normalize( va, norma );           /* 単位ベクトル化 */
  normalize( vb, normb );
  vc[0] = ( norma[0]+normb[0] )/2.0;/* 2等分線を取ると */
  vc[1] = ( norma[1]+normb[1] )/2.0;/* p2 における接線ベクトル */
  mult_factor_vec( vc, va );
  mult_factor_vec( vc, vb );
  p1[0] = p2[0]-(int)va[0];         /* p2 の隣2つの制御点 */
  p1[1] = p2[1]-(int)va[1];
  p3[0] = p2[0]+(int)vb[0];
  p3[1] = p2[1]+(int)vb[1];
  return;
}

int       bezier( s1, s2, s3, s4 )  /*4つの制御点から Bezier 曲線を発生する*/
ivector  s1, s2, s3, s4;
{
  ivector  s12, s23, s34;
  double  lx, ly;
#define  s1234  s23           /*再帰が深くなるので制御点を使い回して節約する*/
#define  s123  s2
#define  s234  s3
  lx=(double)(s4[0]-s1[0]);  /* 制御点間の距離が十分短くなったら */
  ly=(double)(s4[1]-s1[1]);  /* 再帰を打ち切る */
  if ( (lx*lx+ly*ly)<((double)MIN_LENGTH*(double)MIN_LENGTH) ) {
    if ( ++N_pts2>Maxpts2 ) return ( 1 );
    SCPY( pts2[N_pts2], s2 );/* 曲線を微小線分で近似する */
    if ( ++N_pts2>Maxpts2 ) return ( 1 );
    SCPY( pts2[N_pts2], s3 );
    if ( ++N_pts2>Maxpts2 ) return ( 1 );
    SCPY( pts2[N_pts2], s4 );
    return ( 0 );
  }

#define  MID( V1, V2, V12 )   { V12[0]=(V1[0]+V2[0])/2; V12[1]=(V1[1]+V2[1])/2; }
  MID( s1, s2, s12 );         /* 中点を取っていく */
  MID( s2, s3, s23 );
  MID( s3, s4, s34 );
  MID( s12, s23, s123 );
  MID( s23, s34, s234 );
  MID( s123, s234, s1234 );
  if ( bezier( s1, s12, s123, s1234 )!=0 ) return ( 1 );/*再帰分割*/
  if ( bezier( s1234, s234, s34, s4 )!=0 ) return ( 1 );/*出力点列が不足すればエラー*/

#undef  s1234
#undef  s123
#undef  s234

  return ( 0 );

}

extern unsigned char  OVERSAMPLE_NOTYET[];
unsigned char  CURVE_TOO_MANY[]="入力点の数が多すぎます";
unsigned char  CURVE_EXHAUSTED[]="出力の配列の大きさが足りません";

FUNC  pts_curve( dummy )  /* 関数本体 */
DUMMY  dummy;
/* PTS *pts1, int w1, int w2, PTS *pts2 */
{
  PTS  *pts1;
  int  w1, w2;
  int  i, j, e, n1, n2, m;
  ARGSET( dummy );
  ARYSET(1);
  pts1=PARYTOP(1);
  w1=IVALUE(2);
  w2=IVALUE(3);
  ARYSET(4);
  pts2=PARYTOP(4);
  Maxpts2=1;
  for ( i=0; i<DIM(4); i++ ) {
    Maxpts2 *= ( SUFFIX(4,i+1)+1 );
  }
  Maxpts2 /= PTSSIZE;  /* 出力点列の長さの最大値 */

  N_pts1=pts1[0][0];   /* 入力点列の長さ */
  Type_pts=pts1[0][1]; /* 入力点列のタイプ */
  if ( N_pts1>MAXPTS1 ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _CURVE_TOO_MANY,a1" );
#else
#asm
  lea.l  _CURVE_TOO_MANY,a1
#endasm
#endif
    return ( 1 );
  }
  if ( pts1[0][2]!=OVERSAMPLE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _OVERSAMPLE_NOTYET,a1" );
#else
#asm
  lea.l  _OVERSAMPLE_NOTYET,a1
#endasm
#endif
    return ( 1 );
  }
  for ( i=1; i<=N_pts1; i++ ) {
    Ptstemp[i][0][0]=pts1[i][0]*SCALE;  /* サンプル点を制御点 */
    Ptstemp[i][0][1]=pts1[i][1]*SCALE;
  }
  /* 制御点の前処理 … サンプル点間を3等分する */
  for ( i=1; i<N_pts1; i++ ) {
    Ptstemp[i][1][0]=(Ptstemp[i][0][0]*2+Ptstemp[i+1][0][0])/3;
    Ptstemp[i][1][1]=(Ptstemp[i][0][1]*2+Ptstemp[i+1][0][1])/3;
    Ptstemp[i][2][0]=(Ptstemp[i][0][0]+Ptstemp[i+1][0][0]*2)/3;
    Ptstemp[i][2][1]=(Ptstemp[i][0][1]+Ptstemp[i+1][0][1]*2)/3;
  }
  if ( Type_pts==CYCLIC ) {
    Ptstemp[N_pts1][1][0]=(Ptstemp[N_pts1][0][0]*2+Ptstemp[1][0][0])/3;
    Ptstemp[N_pts1][1][1]=(Ptstemp[N_pts1][0][1]*2+Ptstemp[1][0][1])/3;
    Ptstemp[N_pts1][2][0]=(Ptstemp[N_pts1][0][0]+Ptstemp[1][0][0]*2)/3;
    Ptstemp[N_pts1][2][1]=(Ptstemp[N_pts1][0][1]+Ptstemp[1][0][1]*2)/3;
  }
  /* 制御点の発生 */
  for ( i=2; i<N_pts1; i++ ) {
    control( Ptstemp[i-1][2], Ptstemp[i][0], Ptstemp[i][1] );
  }
  if ( Type_pts==CYCLIC ) {
    control( Ptstemp[N_pts1-1][2], Ptstemp[N_pts1][0], Ptstemp[N_pts1][1] );
    control( Ptstemp[N_pts1][2], Ptstemp[1][0], Ptstemp[1][1] );
  }
  /* 自由曲線の生成 */
  SCPY( pts2[1], Ptstemp[1][0] );  /* 始点はマニュアルでコピー */
  if ( w1<=0 || w2<=0 ) {
    m=1;  /* 線の太さを点列レベルで（頂点ごとに）指定されている */
  } else {
    m=0;  /* 線の太さはコマンドレベル（点列全体）で指定されている */
  }
  N_pts2=1;
  for ( i=1; i<N_pts1; i++ ) {
    n1=N_pts2;
    e=bezier( Ptstemp[i][0], Ptstemp[i][1], Ptstemp[i][2], Ptstemp[i+1][0] );
    n2=N_pts2;
    if ( m ) {
      for ( j=n1; j<=n2; j++ ) {
        pts2[j][2]=(pts1[i][2]*(n2-j)+pts1[i+1][2]*(j-n1))/(n2-n1);
      }
    }
  }
```

▶三国志IIよ，14,800円ためて待ってるから早く出ておいで。　神尾　俊暢(16)京都府

```
  if ( e==0 && Type_pts==CYCLIC ) {  /* 点列が循環する場合 */
    n1=N_pts2;
    e=bezier( Ptstemp[N_pts1][0], Ptstemp[N_pts1][1], Ptstemp[N_pts1][2], Ptstemp[1][0] );
    n2=N_pts2;
    N_pts2--;  /* 終点は始点と一致するので捨てる */
    if ( m ) {
      for ( j=n1; j<=n2; j++ ) {
        pts2[j][2]=(pts1[i][2]*(n2-j)+pts1[1][2]*(j-n1))/(n2-n1);
      }
    }
  }
  if ( e!=0 ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _CURVE_EXHAUSTED,a1" );
#else
#asm
  lea.l  _CURVE_EXHAUSTED,a1
#endasm
#endif
    return ( 1 );
  }
  pts2[0][0]=N_pts2;  /* ヘッダをつける */
  pts2[0][1]=Type_pts;
  pts2[0][2]=OVERSAMPLE;
  if ( m==0 ) {
    for ( i=1; i<=N_pts2; i++ ) {  /* 各区間の幅を補間しながら設定する */
      pts2[i][2]=(w1*(N_pts2-i)+w2*(i-1))/(N_pts2-1);
    }
  }
  return ( 0 );
}
```

リスト9

```
================  pts_procs.c  ==================
/****** 点列の移動および接続 ******/

#include  "anti.h"

unsigned char OVERSAMPLE_NOTYET[]="オーバーサンプリング座標に変換してください";

unsigned char MOVE_INCOMPATIBLE[]="移動先の点列とサイズが合いません";

FUNC   pts_move( dummy )
DUMMY  dummy;
/* PTS *pts1, int x, int y, PTS *pts2 */
{
  PTS  *pts1, *pts2;
  int  x, y, i;
  int  n1, n2;

  ARGSET( dummy );
  ARYSET(1);
  pts1=PARYTOP(1);
  x=IVALUE(2);
  y=IVALUE(3);
  ARYSET(4);
  pts2=PARYTOP(4);
  n2=1;
  for ( i=0; i<DIM(4); i++ ) {
    n2 *= ( SUFFIX(4,i+1)+1 );
  }
  n2 /= PTSSIZE;
  if ( pts1[0][2]!=OVERSAMPLE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _OVERSAMPLE_NOTYET,a1" );
#else
#asm
  lea.l  _OVERSAMPLE_NOTYET,a1
#endasm
#endif
    return ( 1 );
  }
  n1=pts1[0][0];
  if ( (n1+1)>n2 ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _MOVE_INCOMPATIBLE,a1" );
#else
#asm
  lea.l  _MOVE_INCOMPATIBLE,a1
#endasm
#endif
    return ( 1 );
  }
  pts2[0][0]=n1;
  pts2[0][1]=pts1[0][1];
  pts2[0][2]=OVERSAMPLE;
  for ( i=1; i<=n1; i++ ) {
    pts2[i][0]=pts1[i][0]+x;
    pts2[i][1]=pts1[i][1]+y;
    pts2[i][2]=pts1[i][2];
  }
  return ( 0 );
}

unsigned char  APPEND_INSUFFICIENT[]="移動先の点列のサイズが足りません";

FUNC  pts_append( dummy )
DUMMY  dummy;
/* PTS *pts1, int x, int y, PTS *pts2 */
{
  PTS  *pts1, *pts2;
  int  x, y, i;
  int  n1, n2;

  ARGSET( dummy );
  ARYSET(1);
  pts1=PARYTOP(1);
  ARYSET(2);
  pts2=PARYTOP(2);
  if ( pts1[0][2]!=OVERSAMPLE || pts2[0][2]!=OVERSAMPLE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _OVERSAMPLE_NOTYET,a1" );
#else
#asm
  lea.l  _OVERSAMPLE_NOTYET,a1
#endasm
#endif
    return ( 1 );
  }
  n1=1;
  for ( i=0; i<DIM(1); i++ ) {
    n1 *= ( SUFFIX(1,i+1)+1 );
  }
  n1 /= PTSSIZE;
  n2=pts2[0][0];
  if ( n1<(pts1[0][0]+n2) ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _APPEND_INSUFFICIENT,a1" );
#else
#asm
  lea.l  _APPEND_INSUFFICIENT,a1
#endasm
#endif
    return ( 1 );
  }
  n1=pts1[0][0];
  pts1[0][0]=n1+n2-1;
  x=pts1[n1][0]-pts2[1][0];  /*pts1 の終点とpts2の始点を一致させる*/
  y=pts1[n1][1]-pts2[1][1];
  for ( i=1; i<n2; i++ ) {
    pts1[i+n1][0]=pts2[i+1][0]+x;
    pts1[i+n1][1]=pts2[i+1][1]+y;
    pts1[i+n1][2]=pts2[i+1][2];
  }
  return ( 0 );
}

unsigned char  OVERSAMPLE_ALREADY[]="オーバーサンプリングずみです";

FUNC  pts_oversample( dummy )
DUMMY  dummy;
/* PTS *pts */
{
  PTS  *pts;
  int  n, i;

  ARGSET( dummy );
  ARYSET(1);
  pts=PARYTOP(1);
  if ( pts[0][2]==OVERSAMPLE ) {
#ifdef  __GNUC__
    asm ( "  lea.l  _OVERSAMPLE_ALREADY,a1" );
#else
#asm
  lea.l  _OVERSAMPLE_ALREADY,a1
#endasm
#endif
    return ( 1 );
  }
  pts[0][2]=OVERSAMPLE;
  n=pts[0][0];
  for ( i=1; i<=n; i++ ) {
    pts[i][0] = OVER( pts[i][0] );
    pts[i][1] = OVER( pts[i][1] );
  }
  return ( 0 );
}
```

リスト10

```
=================  anti.h  ==================
/****** 汎用マクロなどの定義 ******/

#define     PTSSIZE     3  /* 輪郭を点列で表現する */
typedef int PTS[ PTSSIZE ];

#define    N_PIXEL    512  /* スクリーンのサイズは 512×512 ピクセル */

#define    OVERSAMPLE  8   /* オーバーサンプリング倍数 */
#define    OVER2 (OVERSAMPLE*OVERSAMPLE) /*1ピクセルあたりのサブピクセル数*/

/* 通常の座標からオーバーサンプリング座標に変換する、など */
#define    OVER( X )   ((X)*OVERSAMPLE+(OVERSAMPLE/2))
#define    PIX( X )    ((X)/OVERSAMPLE)
#define    SUBPIX( X ) ((X)%OVERSAMPLE)

/* 点列のフォーマット **************

PTS  *pts;  で宣言してあるとする

  pts[0][0]  点列を構成する点の数
  pts[0][1]  点列のタイプ (片道通行か循環しているか)
  pts[0][2]  オーバーサンプリング倍数 (ここが OVERSAMPLE でないなら描画関数はエラーになる)

** 第 i 点の情報 ( 1≦i≦pts[0][0] ) **

  pts[i][0]  x 座標
  pts[i][1]  y 座標
  pts[i][2]  線の幅 (この値が OVERSAMPLE なら 1ピクセルぶんの幅)

**********************************/

/* 点列のタイプ:片道通行か循環しているのか … pts[0][1] の値 */

#define  NORMAL    0
#define  CYCLIC    1


/* X-BASIC からの引数をアクセスする */

typedef int  FUNC;     /* X-BASIC の外部関数:戻り値はエラーコード */
typedef int  DUMMY;    /* Cとの引数の受け渡しの違いを吸収するダミー引数 */

extern unsigned short  *par;      /* 一時的な引数リスト */
extern unsigned short  *ary[10+1];/* 一時的な配列リスト:X-BASIC の引数は最大 10 個 */

#define  ARGSET( A )  { par=(unsigned short *)(&A); } /*引数*/
#define  ATOP( I )    ( (I)*5-4 )                      /*第 I 引数の先頭*/
#define  TYPE( I )    ( par[ATOP(I)] )                 /*引数の型 */
#define  IVALUE( I )  ( par[ATOP(I)+3]<<16 | par[ATOP(I)+4] ) /*intの値*/
#define  CVALUE( I )  ( par[ATOP(I)+4] )               /*char の値*/
#define  ARYSET( I )  { ary[I]=(unsigned short *)IVALUE(I); } /* 配列 */
#define  DIM( I )     ( ary[I][2]+1 )                  /*配列の次元 */
#define  ELEMENT( I ) ( ary[I][3] )                    /*配列要素のサイズ*/
```


▶なぜかムショーに，X68000で“ザナドゥ”をやってみたい。 北田 秀孝(19)和歌山県

```
#define  SUFFIX( I, J) ( ary[I][J+3] )                    /*第 J 添字の最大値*/
#define  ARYTOP( I )   ( &(ary[I][DIM(I)*3+2]) )          /*配列の先頭*/
#define  IARYTOP( I )  ( (int *)ARYTOP(I) )               /*int  配列の先頭*/
#define  CARYTOP( I )  ( (unsigned char *)ARYTOP(I) )     /*char 配列の先頭*/
#define  PARYTOP( I )  ( (PTS *)ARYTOP(I) )               /*PTS  配列の先頭*/

/* その他、便利なマクロ */

#define  ABS( X )  (((X)>0)?(X):(-(X)))            /* X の絶対値 */
#define  SGN( X )  (((X)>0)?1:(((X)<0)?(-1):0))  /*X の符号 (正負または零) */

#define  MIN( X, Y )  (((X)>(Y))?(Y):(X))    /* X,Y のうち大きくないほう */
#define  MAX( X, Y )  (((X)>(Y))?(X):(Y))    /* X,Y のうち小さくないほう */

/* アンチエリアシング関係 … ペイントおよびソリッドスキャンコンバージョン */

#define    N_TILE  8  /* 格納できるタイルパターンの数 */
#define    N_TONE  8  /* 格納できるトーンパターンの数 */
#define    T_SIZE 64  /* タイル及びトーンパターンの大きさ */

#define    COLOR   0  /* 単色で塗り潰す */
#define    TILE    1  /* タイルパターンにしたがって色をつける */

#define    OFF_NTP 0  /* トーンは使わない・ベタ塗り */
#define    ON_NTP  1  /* トーンを使う    ・ベタ塗り */
#define    OFF_TP  2  /* トーンは使わない・下地は透ける */
#define    ON_TP   3  /* トーンを使う    ・下地は透ける */
#define    ON      1  /* トーンを使う */
#define    TP      2  /* 下地は透ける */
```

```

          /* R,G,B ごとの輝度を得るためのマスクとビットシフト */
#define    VMASK_B    62    /* 0b0000000000111110 */
#define    VMASK_R    1984  /* 0b0000011111000000 */
#define    VMASK_G    63488 /* 0b1111100000000000 */

#define    SHIFT_B    1
#define    SHIFT_R    6
#define    SHIFT_G    11

          /* 輝度ビットの値を取り出すためのマスク */
#define    VMASK_I         1    /* 0b0000000000000001 */
#define    VMASK     VMASK_R    /* 輝度の代表値は赤プレーンから取ってくる */
#define    VSHIFT    SHIFT_R
#define    FMASK     VMASK_I    /* ペイント済みフラグには輝度ビットを用いる */
#define    SLMASK    (FMASK|VMASK)

          /* R,G,B および R,G,B,I からカラーコードを計算する */
#define    RGB(R,G,B)      ((B)<<SHIFT_B|(R)<<SHIFT_R|(G)<<SHIFT_G)
#define    RGBI(R,G,B,I)  ((B)<<SHIFT_B|(R)<<SHIFT_R|(G)<<SHIFT_G|(I))

          /* カラーコードから R,G,B,I 成分を取り出す */
#define    BLUE(C)    ( ((C)&VMASK_B) >> SHIFT_B )
#define    RED(C)     ( ((C)&VMASK_R) >> SHIFT_R )
#define    GREEN(C)   ( ((C)&VMASK_G) >> SHIFT_G )
#define    INTENSITY(C)  ( (C)&VMASK_I )
#define    VALUE(C)   ( ((C)&VMASK) >> VSHIFT )
#define    IMAX  31  /* R,G,B の輝度の最大値 */
```

リスト11

```
==================  anti.s  ==================
****** 外部関数ヘッダ ******
* pts_curve.c
*       pts_curve( PTS *pts1, int w1, int w2, PTS *pts2 )
* pts_procs.c
*       pts_append( PTS *pts1, PTS *pts2 )
*       pts_move( PTS *pts1, int x, int y, PTS *pts2 )
*       pts_oversample( PTS *pts )
* aa_lines.c
*       aa_lines( PTS *pts, int c )
*       lines( PTS *pts, int c )
* aa_scanconv.c
*       aa_scanconv( PTS *pts, int cmode, int c/n_tile, int tmode, int n_tone )
*       scanconv( PTS *pts, int c )
* aa_paint.c
*       aa_paint( int x, int y, int cmode, int c/n_tile, int tmode, int n_tone )
* aa_procs.c
*       tile_get( int n, int x1, int y1, int x2, int y2 )
*       tone_get( int n, int x1, int y1, int x2, int y2 )
*       whitepaper( void )
*       reverse( void )
*       maskclear( void )
*       monotone( void )
* インフォメーション・テーブル
        dc.l    X_INIT
        dc.l    X_RUN
        dc.l    X_END
        dc.l    X_SYS
        dc.l    X_BRK
        dc.l    X_CTRL_D
        dc.l    X_RES1
        dc.l    X_RES2
        dc.l    PTR_TOKEN
        dc.l    PTR_PARAM
        dc.l    PTR_EXEC
        dc.l    0,0,0,0,0
X_INIT:
X_RUN:
X_END:
X_SYS:
X_BRK:
X_CTRL_D:
X_RES1:
X_RES2:
        rts

* 関数名テーブル
PTR_TOKEN:
        dc.b    'pts_curve',0
        dc.b    'pts_append',0
        dc.b    'pts_move',0
        dc.b    'pts_oversample',0
        dc.b    'lines',0
        dc.b    'aa_lines',0
        dc.b    'scanconv',0
        dc.b    'aa_scanconv',0
        dc.b    'aa_paint',0
        dc.b    'tile_get',0
        dc.b    'tone_get',0
        dc.b    'whitepaper',0
        dc.b    'reverse',0
        dc.b    'maskclear',0
        dc.b    'monotone',0
        dc.b    0
        .even

* パラメータ・テーブルへのポインタ

PTR_PARAM:
        dc.l    PTS_CURVE_PAR
        dc.l    PTS_APPEND_PAR
        dc.l    PTS_MOVE_PAR
        dc.l    PTS_OVERSAMPLE_PAR
        dc.l    LINES_PAR
        dc.l    AA_LINES_PAR
        dc.l    SCANCONV_PAR
        dc.l    AA_SCANCONV_PAR
        dc.l    AA_PAINT_PAR
        dc.l    TILE_GET_PAR
        dc.l    TONE_GET_PAR
        dc.l    WHITEPAPER_PAR
        dc.l    REVERSE_PAR
        dc.l    MASKCLEAR_PAR
        dc.l    MONOTONE_PAR

* パラメータ・テーブル

int_val:        equ $0002       * int
PTS_ary:        equ $0052       * 1D-array of PTS ( 2D-array of int )
fic_ary:        equ $0037       * 1D-array of float,int,char
void_ret:       equ $ffff       * void

PTS_CURVE_PAR:
```

```
        dc.w    PTS_ary
        dc.w    int_val
        dc.w    int_val
        dc.w    PTS_ary
        dc.w    void_ret
PTS_APPEND_PAR:
        dc.w    PTS_ary
        dc.w    PTS_ary
        dc.w    void_ret
PTS_MOVE_PAR:
        dc.w    PTS_ary
        dc.w    int_val
        dc.w    int_val
        dc.w    PTS_ary
        dc.w    void_ret
PTS_OVERSAMPLE_PAR:
        dc.w    PTS_ary
        dc.w    void_ret
LINES_PAR:
        dc.w    PTS_ary
        dc.w    int_val
        dc.w    void_ret
AA_LINES_PAR:
        dc.w    PTS_ary
        dc.w    int_val
        dc.w    void_ret
SCANCONV_PAR:
        dc.w    PTS_ary
        dc.w    int_val
        dc.w    void_ret
AA_SCANCONV_PAR:
        dc.w    PTS_ary
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    void_ret
AA_PAINT_PAR:
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    void_ret
TILE_GET_PAR:
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    void_ret
TONE_GET_PAR:
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    int_val
        dc.w    void_ret
WHITEPAPER_PAR:
        dc.w    void_ret
REVERSE_PAR:
        dc.w    void_ret
MASKCLEAR_PAR:
        dc.w    void_ret
MONOTONE_PAR:
        dc.w    void_ret

* 関数へのポインタ
PTR_EXEC:
        dc.l    _pts_curve
        dc.l    _pts_append
        dc.l    _pts_move
        dc.l    _pts_oversample
        dc.l    _lines
        dc.l    _aa_lines
        dc.l    _scanconv
        dc.l    _aa_scanconv
        dc.l    _aa_paint
        dc.l    _tile_get
        dc.l    _tone_get
        dc.l    _whitepaper
        dc.l    _reverse
        dc.l    _maskclear
        dc.l    _monotone
        .even
```

▶７月号を買いに行った。どの本屋にもなかった。はるかかなたの本屋まで行ってやっと買った。急いで帰って読んだ。落丁していた。その本屋までとりかえに行った。……悲しいぞ。
板垣　央(16)千葉県

X-BASICによる画像処理

後処理によるジャギーの除去

Nakano Shuichi 中野 修一

すでに描かれた絵のギザギザした部分を滑らかにする，そんな処理はできないでしょうか(もちろん，ぼかしや手作業じゃなく)。ここでは3通りのアプローチで輪郭線を綺麗にすることを考えてみます。同時にX-BASICでのグラフィック処理の基本から見ていきましょう。

X68000によるグラフィックの扱い

ふつうコンピュータグラフィックというのは，画面上の点の色の集まりに還元される。さらに，テレビなどでは色の基本は緑赤青で作られる。これら光の3原色でだいたいの色は作れるわけだ。

緑＋赤＝黄
緑＋青＝水色（シアン）
赤＋青＝紫（マゼンタ）
緑＋赤＋青＝白

のような具合だ。

X68000では16色，256色，65536色のグラフィック画面を扱える。16は2の4乗，256は2の8乗，65536は2の16乗となる。これらはコンピュータにとってはものすごくきりのいい数字だから，処理も速いしメモリ効率もいい。

これらに対して，標準パレットでは，

GRBI
GGRRRBBB
GGGGGRRRRRBBBBBI

というふうに2進数の各桁が対応している。Gは緑，Rは赤，Bは青，Iがつくとその色が明るくなると思っておけばいい。ついていると1，消えていると0の値をとる。

16色の場合を考えよう。4（2進数で0100という数値）はGRBIのRがついた状態とみなされる。これは暗い赤に相当する。赤と緑を混ぜた明るい黄色なら13（1101）というふうになる。

256色の場合も同様に数値を2進数で表したときの各桁の状況が色の成分を決めている。ただ，256色のときは暗い緑と倍明るい緑があったり，暗い赤，倍明るい赤とその倍明るい赤，暗い青，倍明るい青とその倍明るい青のようになっているだけだ。256色モードでは赤と青を3段階（8階調），緑だけ2段階（4階調）で表すことになっている。

65536色は緑赤青各32階調に明るさが1段階加わったものだ。

要するに色は数字で扱われる。ある数値がどんな色になるかは2進数で表せばわかる。試しに45627という数値を色にしたとき，赤成分はどのようになっているかを見てみよう。BASICから，

?bin$（45627）

とすると，

101100100011101

と答えが出る。下7～11桁の5桁が赤成分だから，01000＝16となる。

このように色をRGB成分に分離して操作することがグラフィック処理の基本となる。

ジャギーをなくす

今回はすでに描いた絵からジャギーを消す，という処理を考えてみたい。

情報量が少ないので完全な処理は理論的に不可能だ。また，ちゃんとした補間をやるととても重いので，以下は補間といっても平均をとっているだけと考えていい。

これをX-BASICで記述するわけだが，処理自体はともかく，今回のプログラムは高速化などはほとんど考えられていないので，内容的にBASICインタプリタ上で動かすのは相当無理がある。処理範囲を狭くして動作チェックを行うのが関の山というところだろう。

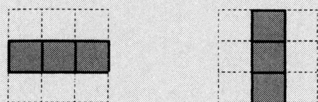
さらにいえば，動作チェックもコンパイルしてからのほうがいい。これならエディタからコンパイラを起動しても変わらないような気もするが，BASICのプログラムには行番号が必要なのに，ED.Xを始めあらゆるエディタにはリナンバー機能がついていない。よってBASICから作業を行うのがもっともよいことになる。

プログラムを直すごとにBASICを抜けてコンパイラを起動するのは面倒なのでチ

表1

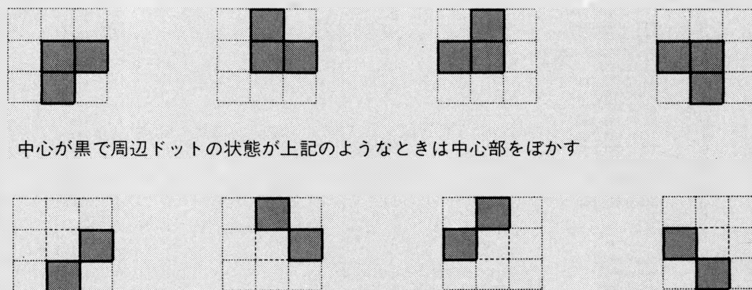
```
key  1,"files @@"
key  2,"load @M"
key  3,"auto "
key  4,"list @M"
key  5,"run @M"
key  6,"/*"
key  7,"width "
key  8,"end"
key  9,"func "
key 10,"system"
key 11,"chdir @@"
key 12,"chdrv @@"
key 13,""
key 14,""
key 15,""
key 16,"sa.@@test@M!gbc test||test@M"
key 17,"sa.@@test@M!ed test.bas@M"
key 18,"img_l@Aoad(@A@I@I@I@I@I)@M"
key 19,""
key 20,""
```

図1



3×3ドットのエリアで見たとき，境界が直線に並んでいると思われる場合は中心のドットは修正しないほうがいい

図2



中心が黒で周辺ドットの状態が上記のようなときは中心部をぼかす

中心が黒以外で周辺ドットが上記のようなときは中心部を軽くぼかす

ャイルドプロセスを使う。さらに，いちいちチャイルドプロセスを起動してコンパイラにたくさんのオプションを与えるのは面倒なので，コンパイラの起動はバッチファイル，BASICからはファンクションキー1発でコンパイル実行できるようにするとよい。

表１のようなファンクションキー設定だとシフト＋F6キーで即座にコンパイル実行できる。RUNコマンドの代わりと思えばいい（メモリの少ない人はできません）。


元画像（協力：高橋哲史）


変換後

輪郭パターンでの補正

まず２月号で行った局所補間つき画面拡大プログラムを見てみよう。これは256×256ドットの絵を512×512ドットに拡大するものだ。ドットをそのまま大きくすると当然モザイクになる。かといって単純に周辺の色と補間して拡大するとボケボケの絵になる。これを防ぐため，輪郭部分を保護しつつ，全体にぼかしをかけることになった(ただし手抜きの処理なので斜め方向は見ていない)。

今回のアンチエリアシング（正確には違うが）でもぼかしを使うことを考えてみよう。絵の輪郭を抽出することは容易だが，そこからベクトルを得ることはちょっと難しいので本格的な処理は私にはできない。

２月号では取り込み画像を対象にしていたため，輪郭保護に重点をおいて明度変化の激しい部分はほっといて，それ以外の部分にぼかしをかけていた。今度はこれとは逆に，明度変化の緩やかな部分は元絵を残し，明度差の激しい部分を選択的にぼかすことになる。しかし，なんでもかんでもぼかすと元絵を大きく損なうので，ぼかさなくてもいい場合を考えよう。

中心が黒でかつ，上下や左右にも黒い点が連続するときはぼかす必要はない(図１)。あまり考えずにアンチエリアシングをやってよさそうなのは，図２に示されるパターンだ，としよう。

まず，輪郭線部分を取り出し，その周辺の状況（輪郭が連続しているかどうか）を配列に読み込む。ある点の周りには８つの点が存在するので，これをビットごとにchar型配列に入れる。

すると256とおりの場合分けができるので，一気にswitchで最適な処理をするというのもいいんだが，ここでは最小限の処理にとどめておく。拡張はご自由に。


拡大するとこうなる

中心点が黒かどうかで図２の上下の処理を選択し，ほぼ全ドットに渡って置き換えを実行する。ぼかしは上下左右のドットの色をRGBごとに重みつきで平均することで行っている。点ごとにだぶった処理を行っているがとりあえず気にしない。これでかなりジャギーが減ったはずだ。

リスト１

```
 10 /*     ---------------------  initialize   ---------*
 20 screen 1,3,1,1
 30 str nam
 40 int g_dat(4,2),col,d(4,2),c(4)
 50 int blue=0,red=1,green=2,i,q=3333
 60 char fl(511,511),fl2(511,511)
 70 /*     ---------------------  main         ---------*
 80 input nam
 90 pic_load(nam+".pic",0,0)
100 edge():beep
110 jag() :beep
120 bokasi()
130 input i
140 end
150 /*     ----------------------------------------------*
160 func edge()
170 for y=1 to 510
180   for x=1 to 510
190     c(0)=point(x,  y  )
200     if c(0)=0 then {
210       fl2(x,y)=1:/*pset(x,y,1)
220 /*    c(1)=point(x+1,y  ) :/*    2    エッジ検出部の名残
230 /*    c(2)=point(x  ,y-1) :/*  4 0 1
240 /*    c(3)=point(x  ,y+1) :/*    3
250 /*    c(4)=point(x-1,y  ) :/*
260 /*    if c(1)>1 or c(2)>1 or c(3)>1 or c(4)>1 then {
270 /*      fl(x,y)=1
280 /*    }
290     } else fl2(x,y)=0
300   next
310 next
320 endfunc
315 /*    ----------------------------------------------*
330 func jag()
340 for y=1 to 510
350   for x=1 to 510
360     col=0
370  /* if fl2(x-1,y-1)=1 then col=col+128
380     if fl2(x  ,y-1)=1 then col=col+64
390. /* if fl2(x+1,y-1)=1 then col=col+32  :/*  7 6 5
400     if fl2(x-1,y  )=1 then col=col+16  :/*  4   3
410     if fl2(x+1,y  )=1 then col=col+8   :/*  2 1 0
420  /* if fl2(x-1,y+1)=1 then col=col+4
430     if fl2(x  ,y+1)=1 then col=col+2
440  /* if fl2(x+1,y+1)=1 then col=col+1
450     fl(x,y)=col
460   next
470 next
480 endfunc
485 /*     ----------------------------------------------*
490 func bokasi()
500 for y=1 to 510            :/*  ぼかし処理
510   for x=1 to 510
520     if fl2(x,y)=1 then {
570       if (fl(x,y) and &B10010)  >16 then fuz(x,y,0)
580       if (fl(x,y) and &B1010)   >8  then fuz(x,y,0)
590       if (fl(x,y) and &B1001000)>64 then fuz(x,y,0)
600       if (fl(x,y) and &B1010000)>64 then fuz(x,y,0)
610     } else {
620       if (fl(x,y) and &B1010000)>64 then fuz(x,y,4)
630       if (fl(x,y) and &B1001000)>64 then fuz(x,y,4)
640       if (fl(x,y) and &B1010)   >8  then fuz(x,y,4)
650       if (fl(x,y) and &B10010)  >16 then fuz(x,y,4)
660     }
670   next
680 next
690 endfunc
695 /*     ----------------------------------------------*
700 func fuz(x,y,p)
710 c(0)=point(x  , y )
720 if c(0)<>1 then {
730   c(1)=point(x,y+1)
740   c(2)=point(x+1,y)
750   c(3)=point(x-1,y)
760   c(4)=point(x,y-1)
770   get_rgb()
780   for m=blue to green
790     d(i,m)=(g_dat(0,m)*p+g_dat(1,m)+g_dat(2,m)+g_dat(3,m)+
g_dat(4,m))¥(p+4)
800   next
810   pset(x,  y,  rgb(d(0,red),d(0,green),d(0,blue)))
820 }
830 endfunc
840 /*     ----------------------------------------------*
850 func get_rgb()
860 for m=blue to green
870   for l=0 to 4
880     g_dat(l,m)=(c(l) mod (1 shl(5*m+6)))shr (m*5+1)
890   next
900 next
910 endfunc
920 /*     ----------------------------------------------*
```


縮小中

ただし，ぼかしを直接画面に描いているので，画面処理されたあとのデータを対象に処理が進んでしまう。これはふつうダサイやり方と呼ばれる。スキャンラインごとに処理をすることもできるので，小さなバッファを取って影響がなくなってから書き込むというのが正しいのだろう。多少処理が複雑になることと，モノがぼかしだけに周りに影響が出ても問題ないんじゃないかという楽観論からこのままにしておいた。

本当は画面分バッファを取って，

int gbuff1(511,511)

のようにしたかったのだが，こういった配列を2つ取ると多くの人のメモリでは収まらないはずなのであきらめた。

それでも512Kバイト分の配列を取っているので，BASIC.CNFを変更してフリーエリアを広げておいてほしい。あとは，PIC.FNC(1990年6月号)をお持ちの方はそのまま，ない方は“pic__”を“img__”に変更して使えばいい。

なお，PIC.FNCを使ったプログラムをコンパイルする場合，ほかのヘッダファイルなどをいじらない限り，パラメータを省略することはできないので注意しよう。ロード先頭座標やセーブする範囲はその都度指定する。当然コンパイル時にはPICLIB.Aも指定すること。

256ドット縮小

さて，アンチエリアシングは悪くいえば不十分なドット数をごまかす手法だ。ふつうはオーバーサンプリングといって「たくさんドットがある」つもりで計算しておき，「実は少なかったんだ」といって1ドットに詰め込むときに平均を取ってやる。

X68000の512×512ドットというのは十分なようで実は少ないともいえる。32ビットマシンなら1024×1024以上が標準だろうし，グラフィックのジャギーを見るとこれくらいはほしくなる。しかし，現状のツールでは真っ正直な線しか考えてない。しかたないからアンチエリアシングするわけだが，現状の512×512ドットのモードがすでにオーバーサンプリングされているとみなすとどうだろうか？

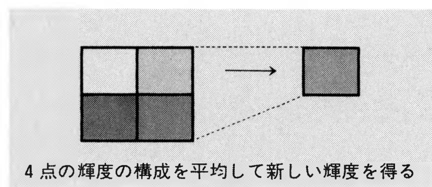
実用上必要なのは綺麗な絵であって高い解像度ではない。なにかとかさむ高解像度の絵より256×256ドットの絵が好まれる場合もある。当然，解像度が低いとジャギーが目立つわけだ。256ドット以下のワンポイント的に使われる絵だって綺麗なほうがいいに決まっている。

となると話は簡単。512×512ドットモードで（当たり前のグラフィックツールを使って）描いた絵を縮小してやればいい。手抜きのグラフィックツールを使うとドットを間引かれるので，ここではプログラムによって平均化された画像を作ることにしよう。

図3のようになった4点をRGB別にして平均し色を決める。小さな画面ならわざわざファイルに書き出したり，大きなバッファを取らなくても画面にそのまま表示できるので結果はリスト2のように単純だ。

ここでは画面の初期化やファイルのロード/セーブを行っていない。BASICで実行しても耐えられない速度ではないということが理由だが，コンパイルして実行したほうがいいに決まっている。必要な人は各自で対応してほしい。また，ファイルのロード時にわざわざinputを使うのも面倒だという場合はコマンドラインから文字を取り込むようにするとよい。Cユーザーズマニュアル参照のこと。

図3


4点の輝度の構成を平均して新しい輝度を得る

リスト2

```
 10 int c(3)
 20 int blue=0,red=1,green=2
 30 int g_dat(3,2),r,g,b
 60 for y=0 to 255
 70   for x=0 to 255
 80     c(0)=point(x*2,y*2)
 90     c(1)=point(x*2+1,y*2)
100     c(2)=point(x*2,y*2+1)
120     c(3)=point(x*2+1,y*2+1)
210     for m=blue to green
220       for l=0 to 3
230         g_dat(l,m)=(c(l) mod (1 shl(5*m+6)))shr (m*5+1)
240       next
250     next
260     b=(g_dat(0,blue)+g_dat(1,blue)+g_dat(2,blue)+g_dat(3,blue))¥4
270     r=(g_dat(0,red )+g_dat(1,red )+g_dat(2,red )+g_dat(3,red ))¥4
280     g=(g_dat(0,green)+g_dat(1,green)+g_dat(2,green)+g_dat(3,green))¥4
290     pset(x,y,rgb(r,g,b))
300   next
310 next
```

1/4補正つき拡大

なにも画像を小さくしなくても，疑似的にオーバーサンプリングできるようにする手もある。簡単にいえば昔使った4倍拡大アルゴリズムで拡大しておいて，今度はそれをモザイク化して1/4の画像を作り出す，という手だ。拡大時に輪郭補正と周りとの平均化を行うので，不正確ながら高解像度のデータを合成することができるだろう。

あとは通常のアンチエリアシングと同様に面積比（といっても4つの平均だが）で色を決定すればいいわけだ。

今回は輪郭色を黒のみに限定して黒のみの補間を行うことにする。それ以外の色ではなにもしない。理由はすぐに縮めるんだからなにもしなくても変わらないからと，黒を残しておけば最初に作ったプログラムをそのまま使ってさらにアンチエリアシングを図ることもできるからだ。

こうしてできたプログラムがリスト3。画面上の256×256の部分を512×512のエリアに拡大する。あらかじめ512×512ドットの絵を1/4ずつに分けてセーブしておいてほしい。

プログラムは同じ画面でもかちあわないように画像の右下から順に処理を進めていく。まず基準点の色を拡大された部分の右隅に打ち，上，左，左上の各ドットの内容から残りの3点の状況を決定する。「両方とも黒ならあいだも黒」というのが基本コンセプトだ。

これだと，左斜めは検出するが，逆の斜めは検出できないので逆斜め専用のループも入れてある。

基準点が黒以外ならなにもしないでその色を4点に置く。このあたりは改良の余地があるかもしれない。

輪郭を黒に限定しない場合なら，単に画

面を1/4ずつに分割して2月号の拡大ルーチンにかけ（自然画でなければ閾値tを多少大きくしたほうがいい），今月の縮小ルーチンで縮めて4つ並べるだけですむ。

労を惜しまず最高のものを得たいなら，適当に下描きした絵を4分割して拡大修正し，また縮小するという手もある。これならふつうのグラフィックツールを使って処理できる。

2Dグラフィックの今後

もともとは取り込み画像に色をつけようとすることから始まった。

まずは高橋哲史君が編集室のスキャナを使って取り込んだ元絵に色をつけようと苦戦している図を想像してもらいたい。ペイントしようとしても途中で止まってしまう。今度はモノクロ2階調で取り込んでペイントしてみる。ちゃんとペイントできるが悲しいくらい絵が粗い。

その場は，2値化して取り込んだ輪郭線を細くして色を塗り，その上に多値化された綺麗な輪郭線を合成する，という方法で落ち着いた。そして，考えられたのが丹氏の多階調境界対応のペイントルーチンだった。

その後，福原君の手作業によるアンチエリアシングを見るにつけ，通常のグラフィックツールの限界と可能性を思い知った。確かに境界線を綺麗に処理してやると非常に高画質な絵が得られることはわかった。しかしそれを手作業で行うというのはあまりに非人間的な作業だろう。これはある程度自動化できそうな処理に思われた。

グラフィックツールはいまだにZ'sSTAFFを最高峰にしたまま進化が止まっている。Z'sSTAFFがよくできたグラフィックツールであることは間違いない。しかし，そろそろもっと凄いものが出てきてもいいんじゃないだろうか。

*　　*　　*

さて，なにはともあれ，必要になるのは十分なメモリだ。たとえばSX-WINDOWでまともなアプリケーションが出てきたとすると，あっというまにメモリが足らなくなるだろう。Macintoshと違いSX-WINDOWは複数のアプリケーションを同時実行することを基本に作られているのでメモリはいくらあっても余ることはない。

考えてみれば，多くの初代X68000やACEユーザーは4万円近く払って1Mバイトの増設を行ったわけだ。それが最近は2MバイトのRAMボードが4万円台で買えるようになってきている。X68000を2,3年も使い込んだユーザーなら，そろそろ増設を考えてもよい頃だろう。効果を考えれば決して高い買い物ではない。

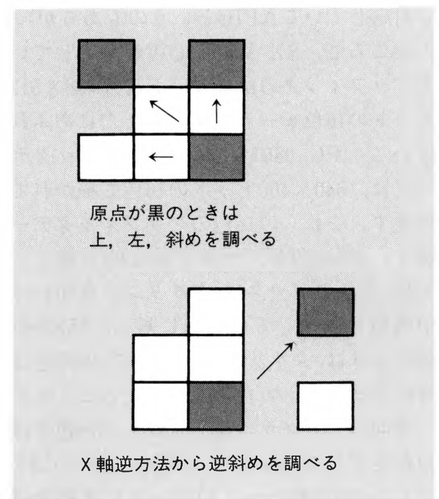
特にグラフィック関係はメモリを大量に必要とする場合が多い。メモリさえあれば内部バッファを1600万色分取って表示部だけ65536色にするなどの方法でより高画質なものを作れる。現状の65536色というのは使っていて極端に不足を感じさせる色数ではない。

グラフィックツールでグラデーションをかけたときやレイトレーシングなどを行ったとき以外不足に感じることはないと思う。どちらも表示関係のルーチンをなんとかすれば，65536色でもかなり自然な表示ができるはずだ。うまくやればグラデの縞模様もマッハバンドも出ない。それは今回鈴木氏の256色化や6月号のSXCONVの16色化を見てもわかると思う。

同様に256色モードでも内部で多色処理すればもっと高度なグラフィックツールができるはずだ。しかし，問題は65536色で描いて変換したほうが綺麗だということだろうか。

65536色モードのデータなら扱いやすくほかのモードへの変換も容易(？)だろう。今後の標準はやはりPIC形式の65536（32768）色となるのだろうか？

図4



リスト3

```
10 screen 1,3,1,1
20 str na
30 int i,j,k(3),l,m,n,o,pl,col(3,2),b,r,g,t=8
40 input na
50 pic_load(na+".pic",0,0)
60 for i=0 to 255
70   for j=0 to 255
80     k(0)=point(255-j  ,255-i  )
90     k(1)=point(255-j  ,255-i-1)
100    k(2)=point(255-j-1,255-i  )
110    k(3)=point(255-j-1,255-i-1)
111    pset(511-j*2,511-i*2,k(0))
120    if k(0)=0 then {
130      if k(1)=0 then pset(511-j*2  ,511-i*2-1,0) else pset(511-j*2  ,511-i*2-1,k(0))
132      if k(2)=0 then pset(511-j*2-1,511-i*2,  0) else pset(511-j*2-1,511-i*2  ,k(0))
134      if k(3)=0 then pset(511-j*2-1,511-i*2-1,0) else pset(511-j*2-1,511-i*2-1,k(0))
135    } else {
140      pset(511-j*2  ,511-i*2-1,k(0))
142      pset(511-j*2-1,511-i*2  ,k(0))
144      pset(511-j*2-1,511-i*2-1,k(0))
149    }
150  next
160  for j=0 to 255
180    k(0)=point(255-j  ,255-i  )
190  /* k(1)=point(255-j  ,255-i-1)
200  /* k(2)=point(255-j+1,255-i  )
210    k(3)=point(255-j+1,255-i-1)
220    if k(0)=0 and k(3)=0 then pset(511-j*2+1,511-i*2-1,0)
330  next
340 next
345 pic_save("ex_"+na,0,0,511,511)
350 input i
360 end
```

色数の補間と量子化

グラフィックデータを変換する

Suzuki Yasuhiro 鈴木 康弘

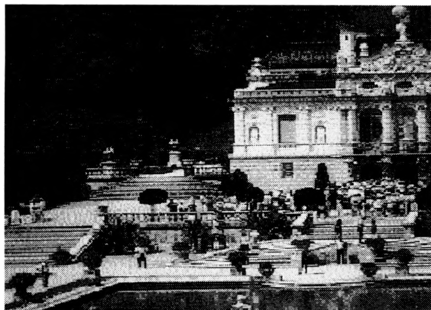
グラフィックモードの違いを埋める処理に挑戦してみましょう。PC-9801などに描かれた16色のグラフィックデータの色数を増やしてX68000の65536色のデータに変換したり，65536色のデータをできるだけ原画に忠実な256色に変換する際に必要な処理を考えてみます。

X68000にはいくつかの種類の画面モードが存在します。そのなかでも，グラフィックにもっとも適しているのは，やはり512×512ドットの65536色モードでしょう。Z's STAFF PRO-68Kが扱うのも，この画面モードですし，PICなどの圧縮ツールもこの画面モード専用です(最近，ほかの画面にも対応しているAPICというのもあるが)。

ところで，SX-WINDOWが対応しているグラフィックの画面モードは，768×512ドットの16色モードです。ちまたにあふれている，PC-9801などのグラフィックデータは，640×400ドットの16色で描かれています。これらの16色のグラフィックデータと，65536色のデータで決定的に違うことは，16色のデータは，タイリングを用いて中間色を表現しているのに対し，65536色のデータは，タイリングを用いず，中間色はそのままドットの色となっていることです。

今回のプログラムは，これらの16色で描かれたグラフィックデータを，512×512ドットの65536色モードのデータに変換を試みたものです。ただし，そのまま変換すると，タイリングされたまま65536色のデータになってしまい(当然65536色中の16色しか使わない)，全然65536色を使っている気分になりません。

また縦横比を調節すると(640×400を512×512に変換するので，1ドットの大きさが変わってくる)，タイリングパターンが崩れてしまい，元のデータよりも汚くなってしまいます。そこで，タイリングで塗られた領域を，なんとかしてそれに対応する色に変換しなければなりません。



オーダードディザ法による変換

逆に色数の多い画面モード用のデータを色数の少ないモード用にコンバートするアルゴリズムは広く知られていますので，それらを使って65536色のデータを256色モードのデータに変換するプログラムも作ってみました。256色モードはグラフィック画面が2枚あり，どうしてもグラフィック画面が1枚では足らないような場合に威力を発揮します。

こっちのほうは，以前Oh!Xで紹介された，オーダードディザ法と，桒野雅彦氏がプリンタのハードコピー用に考え出されたアルゴリズムを応用したものを用いています。また，使われている色数が256色以下の場合は，わざわざディザ法を用いるまでもなく256色モードに変換できるので，その処理を行うこともできます。そのほか，画面中でもっともよく使われている256色を抜き出し，それ以外の色をもっとも近い256色で置き換えるというアルゴリズムも発表されていましたが，今回はそれには対応していません(Oh!X1988年2月号参照)。

ちなみに，65536色に変換するとか書いてありますが，実は32768色に変換します(輝度ビットを無視しています)。また，65536色のデータを256色に変換するのではなく，32768色のデータを256色に変換します(輝度ビットのみ異なる色は，同じ色とみなしています)。

コンパイルの方法

プログラムはC言語で書かれています。したがって，XCが必要になるわけですが，XCでコンパイルされたものはとんでもなく処理速度が遅いのです。そこで，Oh!Xの6月号の特別付録にGCCが掲載されているので，できるだけこっちのほうでコンパイルしてください。

GCCでのコンパイル方法は，

```
gcc T2F.c -O -fstrength-reduce -fomit-frame-pointer -liocs -ldos
gcc to256.c -O -fstrength-reduce -fomit-frame-pointer -liocs -ldos
```

です。ちなみに，XCのほうは，

```
cc T2F.c -O -Y
cc to256.c -O -Y
```

です。

使い方

まず，16色を32768色に変換するT2F.xですが，あらかじめ，画面を16色モードに設定し（実画面のサイズは1024×1024にしてください)，グラフィックデータを表示しておいてください。T2F.xは，VRAMにあるデータを変換します。そして，グラフィックデータが640×400ならば，

```
T2F -S640
```

データが512×512ならば，

```
T2F -S512
```

としてください。これでとりあえず変換を開始します。

また，タイリングパターンを認識して中間色に変換していくので，認識するタイリングパターンの最大値を指定することができます（省略すると，2ドットになります)。たとえば，640×400ドットのデータで，タイリングパターンの最大値を4ドットにするならば，

```
T2F -S640 -T4
```

となります。

この，タイリングパターンの最大値というのは，最大何ドットでタイリングされているか，というものです。たとえば，

黒白白黒白白……

というタイリングがある場合には，最大値に3以上を設定しなければ，これはタイリングとみなされず，そのまま残ってしまいます。

この値をむやみに大きくすると，タイリングでないところまでタイリングとみなしてしまい，変なところが1色で塗られてしまいますから，注意してください。

次に，32768色のデータを256色に変換する，to256.xですが，これもグラフィックを表示させてからプログラムを実行させてください。

使い方は，スイッチに，オーダードディザ法で変換する場合には“-D”,桒野式アルゴリズムで変換する場合には“-K”,色数を数えて，256色以下の絵をそのまま変換する場合には“-C”をつけ加えて起動してください。

オーダードディザ法で変換する場合には，閾値を指定することができます。たとえば，閾値に60を設定したいのなら，

to256 -D60

のように，“-D”に続けて閾値を書きます。省略すると40が設定されます。この値はグラフィックの内容によって最適な値が変わるので，いろいろ試してください。

タイリングについて

16色モードのグラフィックは，ほとんどがタイリングという手法を用いています。このタイリングというのは，たとえば，赤と青のドットを交互に並べていくと，遠目には紫色に見えてしまう，というものです。これを用いると，16色しか出ないはずなのに，それ以上の色を表現することができるのです。

●16色→32768色

まず，タイリングされているグラフィックデータをよ～く見てみますと，タイリングが施されている部分はかなりの規則性があることがわかります。つまり，ある決まったドットの並びが横にず～っと並んでいるのです。色が変わる部分というのは，その決まったドットの並びに合わなくなる部分なのです。

さて，この変換の大まかなアルゴリズムを説明します。

まず，最初にタイリングパターンを横方向に比較していき，そのタイリングパターンが崩れたドットに，フラグを立てて覚えておきます。この処理を全画面に行うと，タイリングパターンが変化した部分（要するに，遠くから見たときの，色が変化する部分→輪郭）にフラグが立つことになります。

あとは，このフラグとフラグのあいだを，その中のタイリングパターンの色で塗っていけばよいのです。

この変換の核となるタイリングパターンが変化した部分の認識ですが，次の手順で行っています。

1) あるドットから右にnドット分を配列変数に格納する
2) さらに，その右nドットが，配列変数に格納した色と同じかどうかを調べる
3) 同じならば，そこからタイリングパターンが続いていることになる
4) 違うのならば,nにn−1を設定して，もう1回調べなおす（1に飛ぶ）
5) nが1になってしまったら，そのドットからはタイリングは始まっていない。したがって，そこにフラグを立てて，1ドット右に移動し,新たに調べ始める（1に飛ぶ）

これで，タイリングパターンが続いているかどうかがわかります。これがわかったら，次はどこまで続いているかです。これは，次々に配列の内容と実際のドットとを比べていき，それらが異なったところまでとなります。

例を出してみると，

座標 0 1 2 3 4 5 6 7 8 9
色　赤 青 赤 青 赤 青 赤 青 黒 黄
10 11 12 13 14
赤 黄 赤 黄 赤……

というドットの並びの場合，まず，先頭の赤青を配列変数に入れます（タイリングの最大値が2の場合）。次に，座標2からの2ドットが，配列変数に入っているものと同じかどうかを調べます。この場合は同じですので，これで赤青というタイリングがあると認識します。

次に，2ドット右に進んで，配列の内容と同じかどうかを調べます。同じですので，さらに2ドット進んで調べます。

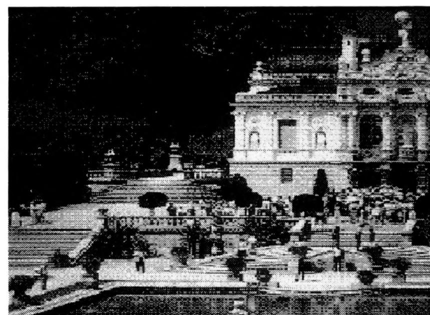
どんどん右に調べていくと，座標8の部分で，配列変数の内容と食い違う色が出てきます。そこで，この座標8のドットにフラグを立てます。

さらに，そこから2ドットを配列に入れます。この場合，黒 黄 が入ります。ところがいきなり次の2ドットと色が異なるため，この黒から始まるのはタイリングではないとみなし，座標9の黄色にフラグを立てます。

そして，次の2ドット（黄 赤）を配列に入れ，再び調べ始めます。

この場合，最初に2ドットを配列に入れて調べましたが，この数値は変更することができ，たとえば3ドットにしてあると，まず3ドットを配列に入れ，そのパターンが続かなければ2ドットにして調べるようになっています。

あと，タイリングパターンから色を決める方法ですが，これは単純に，各タイリングパターンのドットのRGB成分の平均を出し，そのRGBの平均によって作られる色になります。


桒野式アルゴリズムによる変換

2ドットのタイリングでしたら，

求める色＝（色1＋色2）÷2

になります。

●640×400→512×512

実は，今回のプログラムでは，あまりにも色の決定の部分のアルゴリズムの部分に時間をさいてしまい，縦横比の変換はかなりいい加減になっています。したがって，640×400のグラフィックを変換しても，そんなに綺麗にはなりません。

具体的にどうやっているかをばらしますと，まず640×400の32768色が記憶できるバッファを取り，タイリングパターンを調べて色を塗るところまでは，そのバッファに対して処理を行います。その後，画面に512×512で表示する段階になったら，1ドットずつ，対応するドットを調べて，それを画面に表示しているのです。したがって，横方向はところどころドットが抜けて表示され，縦方向はところどころ同じドットが2ドット続きます。

この，抜けたり，2ドット続いたりするのが輪郭の部分だったりすると，輪郭が抜けたり，太くなったりしてしまいます。試しに，640×400のグラフィックでも，512×512で変換してみてください。とりあえず，輪郭は綺麗に変換されると思います。

縦横比を調節すると，輪郭が太くなってしまうというのも欠点ですが，まだあります。

このプログラムでは，横方向しかタイリングを調べていないので，たとえば，

赤白
白白

という，2×2のタイリングパターンには無力です。

実は，このプログラムの最初のバージョンでは，縦方向も調べていたのですが，横方向で調べた輪郭を認識しない部分が出てくるなど，いろいろ問題点が多かったので

す。そこで，これなら縦方向を無視したほうがいいだろうと思い，横方向のみとなったわけです。

●32768色→256色

こちらのプログラムでは，オーダードディザ法と桒野氏のアルゴリズムのどちらかで変換できます。それぞれの詳しい原理などは，以前のOh!Xに載っています。オーダードディザ法は，1988年9月号で丹氏が，桒野式アルゴリズムは，1988年11月号で桒野氏が説明しています（1990年6月号にも掲載されている）から，そちらも参照してください。

オーダードディザ法については電脳倶楽部に最近掲載されたものとアルゴリズムから参考文献まで同じですので，同様の実行結果になるようです。

桒野式（もしかしたら，これが誤差拡散法なのだろうか?)は画面の情報量を減らさずに色変換をする優れたアルゴリズムです。たいていの場合,ディザ法を使うよりも自然な仕上がりになるようです。

RBGという分け方で見る限り，256色というのは半端な値なのですが，ここでは6月号のSXCONV（これは65536色を16色に変換する）と同様な考え方に基づき，

GGGRRRBB

と，もっとも輝度の低そうな青成分を2ビット，ほかを3ビットで処理することによって自然な色に変換しています。ちなみにX68000標準のパレット設定では，

GGRRRBBB

のように，緑が2ビットで処理されています。

色数が256色以下のグラフィックについては，パレットを変更することによりそのままの画像で256色モードに変換できます。256色しか使われていないグラフィックは少ないように思えるかもしれませんが，レイトレや取り込みなどを除く，人が描いたようなグラフィックでしたら，たいていの場合256色以下しか使われていません。

おわりに

最初は画面全体にボカシをかけて，色が急に変化する部分を見つけ出し，そこを輪郭として色を塗っていく，という路線で作っていましたが，どうもうまく輪郭が認識できませんでした。

256色に変換するというのも，使われている色数が256色以上の場合は，似た色を同じパレットに割り当てる，という路線で攻めていましたが，いまいち実行速度が遅くなります。なんとかして高速化を図ろうとしたのですが，いつのまにかオーダードディザ法と桒野式アルゴリズムに落ち着いてしまいました。

今回はええいくそ，という掛け声とともに削除されたファイルが数知れず（その直後に，しまった，という掛け声とともに復活されたファイルも数知れず)，なのでした。

リスト1

```
==================  T2F.c  ==================
/*                                                          */
/*        １６色→３２７６８色 コンバータ  version 1.20       */
/*                                                          */
/*                          by Yasuhiro Suzuki              */
/*                                                          */
/*                                                          */

#include <stdio.h>
#include <stdlib.h>
#include <doslib.h>
#include <iocslib.h>

#define ushort  unsigned short
#define uint    unsigned int


/*****************************/
/*  グローバル変数の宣言  */
/*****************************/
ushort  *buf;                   /* バッファへのポインタ */
int     xsize = 640;            /* 元の絵のＸ座標のドット数 */
int     ysize = 400;            /* 元の絵のＹ座標のドット数 */
int     tmax = 2;               /* 識別するタイリングの最大 ット数 */
int     pg[16];                 /* 元の絵のパレット（Ｇ） */
int     pr[16];                 /* 元の絵のパレット（Ｒ） */
int     pb[16];                 /* 元の絵のパレット（Ｂ） */


/*****************************/
/*  バッファを初期化する  */
/*****************************/
void bclr()
{
    ushort  *p = buf;
    int     i;

    for( i=xsize*ysize; i>0; i-- ){          /* １で初期化しているの
は、*/
        *(p++) = 1;                          /* 黒（０）と区別するた
め */
    }
}

/*************************************/
/*  パレットをＲＧＢに分解して保存  */
/*************************************/
void ptrns()
{
    ushort  *p = (ushort *)0xE82000;         /* パレットの先頭アドレ
ス */
    int     i, c;

    for( i=0; i<16; i++ ){
        c = *(p++);
        pg[i] = ( c >> 11 ) & 0x1F;
        pr[i] = ( c >>  6 ) & 0x1F;
        pb[i] = ( c >>  1 ) & 0x1F;
    }
}

/***********************************/
/*  コマンドオプションを調べる  */
/***********************************/
int chksw( ac, av )
int     ac;
char    *av[];
{
    int     i, c;

    for( i=1; i<ac; i++ ){
        if(( av[i][0] == '-' ) || ( av[i][0] == '/' )){
            c = av[i][1] | 0x20;
            if( c == 's' ){                 /* S */
                xsize = atoi( &av[i][2] );
                if( xsize == 512 ){
                    ysize = 512;
                }
                else if( xsize == 640 ){
                    ysize = 400;
                }
                else{
                    return( 1 );
                }
            }
            else if( c == 't' ){            /* T */
                tmax = atoi( &av[i][2] );
                if( tmax == 0 ){
                    return( 1 );
                }
            }
            else{
                return( 1 );
            }
        }
        else{
            return( 1 );
        }
    }

    return( 0 );
}

/*************************************/
/*  タイリングのドット数を調べる  */
/*************************************/
int tlen( vp, max )
ushort  *vp;
int     max;
{
    ushort  *p, ti[256];
    int     i, j, r;

    if( max > tmax )
        max = tmax;

    for( i=max; i>1; i-- ){
        p = vp;
        for( j=0; j<i; j++ ){           /* 配列変数に読み込む */
            ti[j] = *(p++);
        }
        r = 1;
        for( j=0; j<i; j++ ){           /* 配列変数と等しいか調べる */
            if( ti[j] != *(p++) ){
                r = 0;
                break;
            }
        }
        if( r ){
            return( i );
        }
    }

    return( 0 );
}

/***********************/
/*  境界色を書き込む  */
/***********************/
void tset( x, y, ti, n )
int     x, y, n;
ushort  *ti;
{
    int     i;
```

```
    uint    g, r, b, c;

    g = r = b = 0;
    for( i=0; i<n; i++, ti++ ){
        g += pg[*ti];
        r += pr[*ti];
        b += pb[*ti];
    }

    g /= n;
    r /= n;
    b /= n;
    c = ( g << 11 ) | ( r << 6 ) | ( b << 1 );

    *( buf + (int)( x + y * xsize )) = c;
}

/*************************/
/*  タイリングを調べる  */
/*************************/
void tilex()
{
    ushort  ti[256];
    ushort  *vp, *vvp;
    int     x, y, t, i;

    vvp = (ushort *)0xC00000;
    for( y=0; y<ysize; y++ ){
        t = 0;
        vp = vvp;
        vvp += 1024;
        for( x=0; x<xsize; ){
            if( t == 0 ){
                if( t = tlen( vp, ( xsize - x ) / 2 ) ){
                    for( i=0; i<t; i++ ){
                        ti[i] = *(vp++);
                    }
                    tset( x, y, ti, t );
                    x += t;
                }
                else{
                    ti[0] = *(vp++);
                    tset( x++, y, ti, 1 );
                }
            }
            else{
                if( t > ( xsize - x ) ){
                    t = xsize - x;
                }
                for( i=0; i<t; i++, x++, vp++ ){
                    if( ti[i] != *vp ){
                        t = 0;
                        break;
                    }
                }
            }
        }
    }
}

/***********************/
/*  境界色で塗り潰す  */
/***********************/
void fullx()
{
    ushort  *bp, *vp, c;
    int     x, y, xx, yy;

    bp = buf;
    for( y=0; y<ysize; y++ ){
        c = *bp;
        for( x=0; x<xsize; x++ ){
            if( *bp != 1 ){
                c = *(bp++);
            }
            else{
                *(bp++) = c;
            }
        }
    }

    vp = (ushort *)0xC00000;
    for( y=0; y<512; y++ ){
        yy = (( y * ysize ) / 512 ) * xsize;
        for( x=0; x<512; x++ ){
            xx = ( x * xsize ) / 512;
            *(vp++) = *( buf + (int)( xx + yy ) );
        }
    }
}

/*********************/
/*  メインルーチン  */
/*********************/
int main( ac, av )
int     ac;
char    *av[];
{
    puts("TILE to FULL ver1.20  by Yasuhiro Suzuki");

    if( chksw( ac, av ) ){
        puts("[使用法] T2F [<スイッチ>] ･･･");
        puts("¥t-S640¥t６４０×４００ドットの絵を変換する。");
        puts("¥t-S512¥t５１２×５１２ドットの絵を変換する。");
        puts("¥t-Tn¥t識別するタイリングパターンのドット数の最大値");
        return( 1 );
    }

    if(( buf = (ushort *)MALLOC( xsize * ysize * 2 )) >= (ushort *)0
x80000000 ){
        puts("メモリが足りません。");
        return( 1 );
    }

    SUPER(0);                   /* スーパーバイザモードになる */

    bclr();                     /* バッファを初期化する */
    ptrns();                    /* パレットを保存 */

    tilex();                    /* タイリングの変化点を摘出する */

    CRTMOD( 12 );               /* 画面を初期化する */
    G_CLR_ON();

    fullx();                    /* ＶＲＡＭに表示する */

    return( 0 );
}
```

リスト2

```
==================  to256.c  ==================
/*                                                              */
/*          ６５５３６色→２５６色 コンバータ  version 2.13      */
/*                                                              */
/*                          by Yasuhiro Suzuki                  */
/*                                                              */
/*                                                              */

#include <stdio.h>
#include <stdlib.h>
#include <iocslib.h>
#include <doslib.h>

#define uchar   unsigned char
#define ushort  unsigned short

#define VRAM    (ushort *)0xC00000


/****************************/
/*・グローバル変数の宣言  */
/****************************/
uchar   *buf;
ushort  pcnv[32768];
int     dn;                         /* しきい値 */
int     bg[2][512];
int     br[2][512];
int     bb[2][512];
int     mat[8][8] = {               /* ディザ法で使う行列 */
      0, 32,  8, 40,  2, 34, 10, 42,
     48, 16, 56, 24, 50, 18, 58, 26,
     12, 44,  4, 36, 14, 46,  6, 38,
     60, 28, 52, 20, 62, 30, 54, 22,
      3, 35, 11, 43,  1, 33,  9, 41,
     51, 19, 59, 27, 49, 17, 57, 25,
     15, 47,  7, 39, 13, 45,  5, 37,
     63, 31, 55, 23, 61, 29, 53, 21
};

/***************/
/*  パレット  */
/***************/
#define IM (256*31)

int         rrr[7]={ IM/7, IM*2/7, IM*3/7, IM*4/7, IM*5/7, IM*6/7,
IM*7/7 };
int         ggg[7]={ IM/7, IM*2/7, IM*3/7, IM*4/7, IM*5/7, IM*6/7,
IM*7/7 };
int         bbb[3]={ IM/3, IM*2/3, IM*3/3 };

unsigned short      rgb[8][8][4];

#define     RMAX    31
#define     GMAX    31
#define     BMAX    31

#define     PALRGB(R,G,B)   ( (((G)*GMAX/7)&31)<<11 | (((R)*RMAX/7)
&31)<<6 | (((B)*BMAX/3)&31)<<1 )

/*******************/
/*  色数を数える  */
/*******************/
int count()
{
    ushort  *vp, c;
    int     i, k;

    for( i=0; i<32768; i++ ){
        pcnv[i] = 0;
    }

    vp = VRAM;
    k = 1;
    for( i=512*512; i>0; i-- ){
        c = ( *(vp++) >> 1 );
        if( pcnv[c] == 0 ){
            pcnv[c] = k++;
        }
    }

    return( k - 1 );
}

/**********************************************/
/*  ＶＲＡＭの内容を２５６色に変換して格納  */
/**********************************************/
void trns()
{
    uchar   *bp;
    ushort  *vp;
    int     i;

    bp = buf;
    vp = VRAM;
    for( i=512*512; i>0; i-- ){
        *(bp++) = pcnv[ *(vp++) >> 1 ] - 1;
    }
}

/**************************************/
/*  オーダードディザで色を変換する  */
/**************************************/
void dither()
{
    uchar   *bp;
    ushort  *vp, c;
```

```
    int     g, r, b, d;
    int     x, y;

    bp = buf;
    vp = VRAM;
    for( y=0; y<512; y++ ){
        for( x=0; x<512; x++ ){
            c = *(vp++);
            g = (( c >> 11 ) & 0x1F ) << 3;
            r = (( c >>  6 ) & 0x1F ) << 3;
            b = (( c >>  1 ) & 0x1F ) << 3;
            d = mat[ y & 0x07 ][ x & 0x07 ];
            g = ( g + d ) / dn / 2;
            r = ( r + d ) / dn;
            b = ( b + d ) / dn;
            if( g > 3 ){
                g = 3;
            }
            if( r > 7 ){
                r = 7;
            }
            if( b > 7 ){
                b = 7;
            }
            *(bp++) = ( g << 6 ) | ( r << 3 ) | b;
        }
    }
}

/*******************************/
/*  桑野式アルゴリズムで変換  */
/*******************************/
void kuwano()
{
    uchar   *bp;
    ushort  *vp, c;
    int     x, y, lc, lb;
    unsigned int     cg, cr, cb, dg, dr, db;
    int     i;

    bp = buf;
    vp = VRAM;
    for( x=0; x<512; x++ ){
        bg[0][x] = br[0][x] = bb[0][x] = 0;
    }
    for( y=0; y<512; y++ ){
        lc = y & 1;
        lb = ( y + 1 ) & 1;
        for( x=0; x<512; x++ ){
            bg[lb][x] = br[lb][x] = bb[lb][x] = 0;
        }
        cg = cr = cb = 0;
        for( x=0; x<512; x++ ){
            c = *vp;
            cg += (( c >> 11 ) & 0x1F ) * 256 + bg[lc][x];
            cr += (( c >>  6 ) & 0x1F ) * 256 + br[lc][x];
            cb += (( c >>  1 ) & 0x1F ) * 256 + bb[lc][x];

            dg=dr=db=0;
            for ( i=6; i>=0; i-- ) {
                if( cg >= ggg[i] ){
                    dg = i+1;
                    cg -= ggg[i];
                    break;
                }
            }
            for ( i=6; i>=0; i-- ) {
                if( cr >= rrr[i] ){
                    dr = i+1;
                    cr -= rrr[i];
                    break;
                }
            }
            for ( i=2; i>=0; i-- ) {
                if( cb >= bbb[i] ){
                    db = i+1;
                    cb -= bbb[i];
                    break;
                }
            }

            *(bp++) = ( dg << 5 ) | ( dr << 2 ) | db;
            *(vp++) = PALRGB( dr, dg, db );

            bg[lb][x] += cg/8;
            br[lb][x] += cr/8;
            bb[lb][x] += cb/8;
            bg[lb][ ( x > 0 )? ( x - 1 ): ( x ) ] += (cg/4);
            br[lb][ ( x > 0 )? ( x - 1 ): ( x ) ] += (cr/4);
            bb[lb][ ( x > 0 )? ( x - 1 ): ( x ) ] += (cb/4);
            bg[lb][ ( x < 511 )? ( x + 1 ): ( x ) ] += cg/8;
            br[lb][ ( x < 511 )? ( x + 1 ): ( x ) ] += cr/8;
            bb[lb][ ( x < 511 )? ( x + 1 ): ( x ) ] += cb/8;
        cg/=2;
        cr/=2;
        cb/=2;
        }
    }
}

/*********************/
/*  パレットを設定  */
/*********************/
void setpal()
{
    ushort  *pp, c;
    int     i;

    pp = (ushort *)0xE82000;
    for( i=0; i<32768; i++ ){
        if(( c = pcnv[i] ) != 0 ){
            *(pp + c - 1 ) = i << 1;
        }
    }
}

/*************************************/
/*  バッファの内容をVRAMに転送  */
/*************************************/
void prt()
{
    uchar   *bp;
    ushort  *vp;
    int     i;

    bp = buf;
    vp = VRAM;
    for( i=512*512; i>0; i-- ){
        *(vp++) = (ushort)*(bp++);
    }
}

/**********************/
/*  画面モード初期化  */
/**********************/
void ginit0()
{
    CRTMOD( 8 );
    G_CLR_ON();
}

void ginit1()
{
    int    r, g, b;

    CRTMOD( 8 );
    G_CLR_ON();

    for ( g=0; g<8; g++ ) {
        for ( r=0; r<8; r++ ) {
            for ( b=0; b<4; b++ ) {
                rgb[r][g][b]=(g<<13|r<<8|b<<4);
                GPALET( (g<<5|r<<2|b), PALRGB( r, g, b ) );
            }
        }
    }
    return;
}

/**********************************/
/*  コマンドオプションをチェック  */
/**********************************/
int chksw( ac, av )
int     ac;
char    *av[];
{
    int     c, i, r=0;

    if( ac < 2 ){
        return( 0 );
    }

    for( i=1; i<ac; i++ ){
        if( av[i][0] != '-' ){
            return( 0 );
        }
        c = av[i][1] | 0x20;
        switch( c ){
            case 'd':   r |= 0x01;
                        dn = atoi( &av[i][2] );
                        if( dn == 0 ){
                            dn = 40;
                        }
                        break;
            case 'k':   r |= 0x02;
                        break;
            case 'c':   r |= 0x04;
                        break;
            default:    return( 0 );
        }
    }

    if((( r & 0x03 ) == 0x03 ) || (( r & 0x03 ) == 0x00 )){
        return( 0 );
    }

    return( r );
}

/*******************/
/*  メインルーチン  */
/*******************/
int main( ac, av )
int     ac;
char    *av[];
{
    int     m;

    puts("65536 to 256 ver2.13  by Yasuhiro Suzuki");

    if(( buf = (uchar *)MALLOC( 512 * 512 )) >= (uchar *)0x80000000
){
        puts("メモリが足りません。");
        return( 1 );
    }

    if( !( m = chksw( ac, av )) ){
        puts("[使用法] to256 <スイッチ>");
        puts("¥t-Dn¥tオーダードディザ法で変換を行う(nはしきい値)
");
        puts("¥t-K¥t桑野式変換を行う");
        puts("¥t-C¥t色数を調べて変換する");
        return( 0 );
    }

    SUPER(0);

    if(( m & 0x04 ) && ( count() <= 256 )){
        trns();
        ginit0();
        setpal();
    }
    else if( m & 0x01 ){
        dither();
        ginit0();
    }
    else{
        kuwano();
        ginit1();
    }

    prt();

    return( 0 );
}
```

4096色→8色変換

Zの画像をX1で

Kameda Masahiko 亀田 雅彦

画像変換処理のX1シリーズでの応用例です。X1turboZの4096色画像を「来野式アルゴリズム」で8色のデータに変換してみましょう。同時にKAME-DOS上でグラフィックを扱うためのコマンドについても解説します。今回のINTEGRAL X1の連載記事もあわせてご覧ください。

なぜ，8色なの？

今月は大盤振る舞いなのです。まさに「もってけどろぼう！」の世界といえるでしょう。なぜかというと，この特集とKAME-DOS連載の豪華2本立てだからです。しかも，それらが見事に調和を保ちながらダブル進行していくという華麗さ，名づけて「シンクロ原稿」です。「ライターがX1関係で荒稼ぎをしようとしてる」とか，「1本のプログラムを使い回してるだけだ」という噂の真偽はさておき，特集とは名ばかり，KAME-DOS関係の話が割り込んでくるので悪しからず。

しかしながら，グラフィック特集である以上グラフィックにも力をいれなければなりません。そこで今回は「Zの4096色画像を8色に変換してみよう」ということになりました。ここでふと思い浮かんでくるのは，6月号のSX-WINDOWのグラフィックについて。パラパラとめくってみると，そのものずばり載っているじゃあないですか。しかもその6月号ですら，1988年11月号の引用なのだから，私は「引用の引用」をするという，神をも恐れぬワザにでようというわけです。でも楽なことはいいことなので，そのまま採用させてもらいました（実際の実行結果も良好でした）。

それじゃ8色に変換してうれしいこと。

●メモリが節約できる

96K（4096色フル）だと2Dディスクで3枚ちょっと。2HD（アクセスが遅い）なら10枚くらいで，結構邪魔です。ディスクアクセス側の問題もありますが，容量はロード時間にも影響を与えます。

●互換性が出てくる

4096色というのはあまりメジャーな数字ではないですが，8,16色あたりはMS-DOSの世界では常識です。もちろんX1のVRAMデータ形式のままでは互換性はありませんが，変換自体は簡単にできそうなので挑戦してみるのも面白そうです。

●プリンタとの相性がいい

実はこれが一番身近な問題だと思います。Zではアナログ画像取り込みが標準で装備されながら，あまり活用されないのはグラフィックの扱いにくさが原因でしょう。カラーイメージボードは8色でありながら，そのデータの少なさがよいほうへ働いています。Z標準のアナログ画像を精一杯有効に活用していきたいとすれば，8色に落としてプリンタへの出力を容易にするのが効果的です。ひょっとすると，安価なスキャナとしての価値をZに見出せるかもしれません。

もちろん，8色にして悪いことは原画の情報がRGBの各色について1/4ずつになることです。これをなるべく緩和しようとするのが前述のアルゴリズムです。

プログラムは？

画像変換のためのものと画面ローダ，画面セーバの3本を用意しました。ローダとセーバに関しては，連載のKAME-DOSの外部コマンドとしても使えるようになっています（そっちがメインだったりして）。もともとKAME-DOSのほうで外部コマンドの許容範囲が広いので，画像変換プログラムもコマンドとすることができます（あまり意味はありません）。

画像変換プログラムは人のアルゴリズムを使っているのであまり自慢できたものじゃありませんが，ローダとセーバのスピードに関しては自信を持っています。KAME-DOSの実力をいかんなく発揮させて，理論的な最高速に達しました。画面全体を一度にロードしたりセーブしたりしかできませんが，そのスピードは一度見てもらえればわかります。

こんなスピードを競うようなプログラムは最近では見かけませんが，8ビット全盛の頃はよくはやったものです（特にグラフィック命令）。自分で書いててなつかしくなりました。

入力方法

●画像変換プログラム（リスト1&リスト2）

X1turboZでなおかつZ-BASIC専用プログラムです（必ずしもKAME-DOSは必要ありません）。Z-BASICからリスト1を入力したら，ファイル名はとりあえず「CCHANGE.X1」としてセーブしておいてください。次に，CLEAR &HC000を実行して，リスト2をなんらかのマシン語入力ツールから打ち込んでください。間違いがなければSAVEM "CCHANGE.OBJ",&HC000,&HC1A7としてセーブします。使うときは両方必要になるので，2つは同一ディレクトリ上においてください。

●画面ローダ

●画面セーバ

X1全シリーズで使うことができます。ただし，6月号から今月にかけて連載しているKAME-DOSシステムが必要になります。具体的には，「INTEGRAL.X」「COMMAND.X1」「FDC.OBJ」の3つのプログラムと，ノーマルX1には7月号のプログラムも必要です。まだ持っていない方は，バックナンバーなどからぜひ入手してください。

「CZ-8FB01,turboBASIC,Z-BASIC」のうちKAME-DOSのあるBASICで，今月の92ページから連載に載っているリストを入力します。「COMMAND.X1」と同一ディレクトリ上にセーブしてください。変数名の間違いがあったりすると，ディスクを破壊しかねないので慎重にチェックしてください。

また，入力上の注意は今月号の連載の「外部コマンド」の入力法のところをよく読んで必ず守るようにしてください。ファイル名はそれぞれ「GLOAD.X1」「GSAVE.X1」とします。

●まとめ

1：Z-BASIC&KAME-DOS

「COMMAND.X1」「CCHANGE.X1」「CCHANGE.OBJ」「GLOAD.X1」「GSAVE.X1」を同一ディレクトリ上においてください。

2：Z-BASICのみの方

「CCHANGE.X1」「CCHANGE.OBJ」を同一ディレクトリ上においてください。

3：KAME-DOSのみの方

「COMMAND.X1」「GLOAD.X1」「GSAVE.X1」を同一ディレクトリ上においてください。

使い方

●CCHANGE.X1（KAME-DOSなし）

あらかじめグラフィックを表示させておいてCCHANGE.X1を起動します。メニュー画面になるので，1を押すと全画面に対して（少し時間がかかりますが）4096色から8色へ変換します。それが終わると，キー入力待ちになって，入力するとメニューへ戻ります。メニューの2，3は使えません。4で終了です。

スペースキーでグラフィックのON/OFFができます。

●CCHANGE.X1（KAME-DOSあり）

KAME-DOSのコマンドライン（[X:/]）から「CCHANGE」として起動します。GLOAD.X1,GSAVE.X1があればメニューの2，3が使えます。それぞれ選択すると，ファイル名の入力になるので，ドライブ名を含めてフルパスで指定してください。リターンキーだけを押せば，メニューに戻ります。その後の操作はGLOAD,GSAVEと同じになります。

メニューからグラフィックをロードすることもできますが，あらかじめロードしておきたいこともあります。そういうときは，グラフィックをロードして，「INTEGRAL.X」の中のグラフィックを消すような命令を削ってから，KAME-DOSを起動してください。

●GLOAD.X1

96Kバイトあるいは64Kバイト（自動的に判断する）のグラフィックファイルをロードします。ファイルの拡張子によって画面モードを自動変更するので注意してください（図1）。

KAME-DOSのコマンドラインから「GLOAD ファイル名」として起動します。エラーがなければ，グラフィックを表示してキー入力待ちになるので，キーを押すと親プロセスへもどります。エラーがあればメッセージを表示して実行を中止します。ロードしている最中は少しキャラクタ画面が乱れますが，それが正常なので心配いりません。ノーマルX1では96Kファイルはロードできません。

Z-BASICには標準でVLOAD,VSAVE

INTEGRAL.Xを書き換える

次に挙げる命令を，自分のINTEGRAL.Xで削除してください。ただし，これはグラフィックをあらかじめロードしておいたときのみの処置なので，通常はいつものINTEGRAL.Xを使ってください。

1040行のWIDTH・1050行のCLS 4
1200行のINIT

また，INTEGRAL.XやCOMMAND.X1にあるSCREEN命令はグラフィック画面を見えなくするものなので，必要に応じて入れておいてください。Z-BASICを使う場合は，OPTION SCREEN 4をOPTIONSCREEN 5に換えておきましょう。

リスト1

```
1000 'CCHANGE.X1  Ver 1.0        By Kameda
1010 '
1020 OPTIONSCREEN 4:WIDTH 40,25,0,1:OPTIONSCREEN 5:INIT:DEFINT a-z
1030 DEFUSR0=m_tranr
1040 '
1050 CLEAR &HC000:LOADM "CCHANGE.OBJ"
1060 '----------( MAIN ROUTINE )-----------
1070 '
1080 SCREEN:CLS:fe$(1)="":GOSUB "menu":CLS
1090 ON a GOTO 1100,"load","save",1160:IF a$=CHR$(27) GOTO 1160
1100 '
1110 KLIST 0:CONSOLE 0,25
1120 MEM$(&HC007,8)=MKI$(0)+MKI$(320)+MKI$(0)+MKI$(200)
1130 OPTIONSCREEN 4:INIT:CFLASH 1:PRINT "Wait a moment.":CFLASH 0
1140 CALL &HC000
1150 GOSUB "ending":GOTO 1080
1160 '
1170 CLS:IF proces=0 THEN END
1180 CLEAR &HD000:proces=proces-1:CHAIN proces$(proces)
1190 '----------( LOAD           )-----------
1200 '
1210 LABEL "load"
1220 IF proces=0 THEN 1080
1230 LOCATE 7, 7:PRINT "***  GRAPHIC LOAD  ***"
1240 LOCATE 11,10:COLOR 1:PRINT "[RETURN]: MENU"
1250 LOCATE 5,13:COLOR 6:PRINT "FILE-NAME>";:COLOR 7:INPUT "",fe$(1)
1260 IF fe$(1)="" THEN 1080 ELSE POKE v_wfd0,PEEK(&HF8D6)
1270 proces$(proces)="CCHANGE.X1":proces=proces+1:CHAIN "GLOAD.X1"
1280 '----------( SAVE           )-----------
1290 '
1300 LABEL "save"
1310 IF proces=0 THEN 1080
1320 LOCATE 7, 7:PRINT "***  GRAPHIC SAVE  ***"
1330 LOCATE 11,10:COLOR 1:PRINT "[RETURN]: MENU"
1340 LOCATE 5,13:COLOR 6:PRINT "FILE-NAME>";:COLOR 7:INPUT "",fe$(1)
1350 IF fe$(1)="" THEN 1080 ELSE POKE v_wfd0,PEEK(&HF8D6)
1360 proces$(proces)="CCHANGE.X1":proces=proces+1:CHAIN "GSAVE.X1"
1370 '----------( MENU           )----------
1380 '
1390 LABEL "menu"
1400 LOCATE  3, 5:PRINT "***  4096 -> 8 color change  ***"
1410 LOCATE 10, 8:COLOR 3:PRINT "[1]";:COLOR 7:PRINT "  COLOR CHANGE"
1420 LOCATE 10,10:COLOR 3:PRINT "[2]";:COLOR 7:PRINT "  GRAPHIC LOAD"
1430 LOCATE 10,12:COLOR 3:PRINT "[3]";:COLOR 7:PRINT "  GRAPHIC SAVE"
1440 LOCATE 10,14:COLOR 3:PRINT "[4]";:COLOR 7:PRINT "  END"
1450 LOCATE 14,16:COLOR 5:PRINT "push [1]-[4]"
1460 LOCATE  7,18:COLOR 7:PRINT "[SPACE] : graphic ON & OFF"
1470 k=0:REPEAT:a$=INKEY$:a=VAL(a$)
1480 IF a$=" " THEN IF k THEN SCREEN:k=0 ELSE INIT:k=1
1490 UNTIL (1<=a AND a<=4) OR a$=CHR$(27)
1500 RETURN
1510 '----------( END            )----------
1520 '
1530 LABEL "ending"
1540 CLS:CFLASH 1:PRINT "PUSH SPACE":CFLASH 0
1550 REPEAT:A$=INKEY$:UNTIL A$<>""
1560 CLS:CONSOLE 0,24:KLIST 1:RETURN
```

リスト2 CCHANGE・OBJ

```
C000 C3 16 C0 00 00 00 00 00 : 99
C008 00 40 01 00 00 C8 00 2C : 35
C010 14 2C 00 00 00 00 21 00 : 61
C018 C8 11 01 C8 01 FF 07 AF : 58
C020 77 ED B0 2A 0B C0 22 05 : 30
C028 C0 D9 DD 21 00 CC 21 00 : 84
C030 C8 D9 3A 05 C0 E6 01 28 : AF
C038 08 D9 E5 DD E5 E1 DD E1 : 27
C040 D9 DD E5 E1 5D 54 13 01 : 41
C048 FF 03 AF 77 ED B0 D9 ED : 8B
C050 5B 07 C0 DD 19 19 D9 CD : D7
C058 6B C0 2A 05 C0 23 22 05 : 64
C060 C0 ED 5B 0D C0 B7 ED 52 : CB
C068 38 BF C9 AF 32 0F C0 32 : A2
C070 10 C0 32 11 C0 2A 07 C0 : C4
C078 22 03 C0 CD 8F C0 2A 03 : 2E
---------------------------------
SUM: 6E 21 02 C9 15 0A 0E F0 B594

C080 C0 23 22 03 C0 ED 5B 09 : 19
C088 C0 B7 ED 52 38 ED C9 CD : 71
C090 80 C1 FD 21 0F C0 3E 40 : AC
C098 32 12 C0 CD 02 C1 FD 7E : 0F
C0A0 00 82 D9 86 D9 16 00 FE : CE
C0A8 78 38 04 D6 78 16 01 CD : E6
C0B0 C7 C0 CD 41 C1 FD 23 DD : 53
C0B8 23 D9 23 D9 3A 12 C0 C6 : CA
C0C0 40 32 12 C0 20 D5 C9 5F : 61
C0C8 CB 3F CB 3F CB 3F FD 77 : 92
C0D0 00 7B E6 07 DD 86 00 FD : C8
C0D8 86 00 DD 77 00 2A 03 C0 : C7
C0E0 7C B5 28 0A FD 7E 00 87 : 65
C0E8 DD 86 FD DD 77 FD FD 7E : 2C
C0F0 00 DD 86 03 DD 77 03 FD : BA
C0F8 7E 00 CB 27 CB 27 FD 77 : D6
---------------------------------
SUM: FC 04 AF 47 39 73 09 0E 43E1

C100 00 C9 CD 9E C1 16 00 CD : D8
C108 25 C1 01 D0 1F ED 78 F6 : 31
C110 10 ED 79 CD 25 C1 01 D0 : FA
C118 1F ED 78 E6 EF ED 79 7A : 39
C120 07 07 07 57 C9 E5 CD 32 : 19
C128 C1 01 00 04 09 CD 32 C1 : 8F
C130 E1 C9 4D 44 ED 78 A3 28 : 6B
C138 04 37 CB 12 C9 B7 CB 12 : 75
C140 C9 CD 9E C1 CD 5D C1 01 : E1
C148 D0 1F ED 78 F6 10 ED 79 : C0
C150 CD 5D C1 01 D0 1F ED 78 : 40
C158 E6 EF ED 79 C9 E5 CD 6A : 20
C160 C1 01 00 04 09 CD 6A C1 : C7
C168 E1 C9 4D 44 CB 1A 38 0A : 62
C170 D5 7B 2F ED 58 A3 ED 79 : CD
C178 D1 C9 ED 78 B3 ED 79 C9 : E1
---------------------------------
SUM: 95 B2 80 32 B7 7A CF A3 31E2

C180 2A 03 C0 ED 5B 05 C0 AF : A9
C188 32 F6 FB 06 1D ED 41 CD : 41
C190 07 59 06 1E ED 41 22 13 : E7
C198 C0 7A 32 15 C0 C9 3A 12 : 56
C1A0 C0 2A 13 C0 84 67 3A 15 : F7
C1A8 C0 5F C9                : E8
---------------------------------
SUM: A3 55 CF E6 A9 63 97 B6 5AB4
```

というグラフィック保存用の命令がありますが，GLOAD,GSAVEのデータ形式はそのフォーマットとまったく同じです。したがって，VSAVEによってセーブされたファイルはGLOADでロードできるし，その逆もまたしかりです（違いは「速さ」だけ)。「ベタ書きフォーマット」であまり賢くないのですが，これが標準なのでしかたありません。

●**GSAVE.X1**

グラフィック画面のセーブです。ロードと同じように起動しますが，セーブする前に96Kバイトか64Kバイトにするかを聞いてきます。画像のグラフィックモードにあわせて決定してください。セーブ時もロード時と同じように画面が乱れます。なお，64KファイルはGSAVE独自のものなので，Z-BASICのVLOADではロードできません。

これらのプログラムをKAME-DOS上で使うときには，重要な注意点がひとつあります。よく読んでください。それは，DOSのバッファをG-RAMに設定している場合です（バッファに関しては6，7月号参照のこと)。画像ファイルをセーブしようとしてディスクアクセスすると，バッファがG-RAM上にあるのでグラフィックが破壊されてしまいます。これでは困るので，バッファをほかに移す必要があるのです。

バンクメモリを搭載していればそこにバッファを設定して一件落着なのですが，そうとばかりは限りません。そこでX1に残された最後の領域であるキャラクタ&アトリビュートエリアに，バッファを設定します（そうです！　このおかげでロード/セーブ時に画面が乱れるのです)。これは一時緊急避難的処置なので，これが終わったらすぐに元へ戻してください。具体的な作業は囲みに書いておきます。

必殺！ アルゴリズム

実は，CCHANGE.OBJ（リスト2）は単独でも使用可能なのです。4096色グラフィックを表示させておいて，CLEAR &HC000：LOADM "CCHANGE.OBJ"でマシン語をロードします。その後，CALL &HC000を実行すれば8色に変換してくれます。これを利用すると，Zでないturboで Zのアナログ画像データをロードし（GLOAD)，8色に変換して，アナロググラフィックをそれなりに見ることもできます（データが手に入れば，だけど)。

また，CCHANGE.X1内で，MEM$(&HC007,8)＝MKI$(0)＋MKI$(320)＋MKI$(0)＋MKI$(200)という行があります。このMKI$の中身は順に左上X座標，右下X座標＋1，左上Y座標，右下Y座標＋1になっていて，この矩形領域が変換対象になります。書き換えて実行してみるとよくわかると思います。

さて，CCHANGEルーチンのアルゴリズムはバックナンバーを見てもらうとして，ここではGLOADとGSAVEについて解説します。

それぞれKAME-DOSのディスクアクセスルーチンを使っているわけですが，KAME-DOSには標準のG-RAMロード&セーブルーチンはありません。どうしてるのかというと，G-RAM全体(48Kバイト×2）をバッファとみなして，通常はデータの仲介役のバッファに，最初からデータを入れておいたことにします。

もともとバッファの大きさは4Kバイト単位の可変長なので48Kバイトでも問題はありません。

この方式の長所としては「BASICでも簡単に制御できる・速くなる」などがあって，短所は「ベタ書きフォーマットにしか通用しない・任意の矩形領域は取り扱いできない」などです。そのためZ'STAFFのフォーマットとは互換性がありません。

＊　　＊　　＊

今回は，特集がメインなのか連載がメインなのかわからなくなってしまいました。ただ，Zに関してはマウス・画像取り込みなど手つかずになってしまったのが残念です。CCHANGEなどはちょっとした変更でまだまだ拡張できるプログラムなので，やりたいことを自分でプログラミングしてみましょう。

図1　拡張子と画面モードの関係

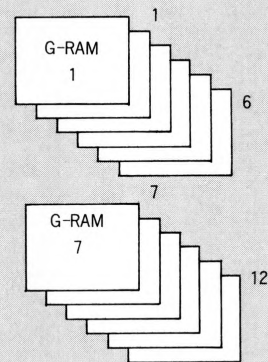
[X:/] GLOAD A：GAZO.GL0

GL0	WIDTH 40,25,0,1	4096色モード
GL1	WIDTH 80,25,0,1	64色モード
GM0	WIDTH 40,25,0,2	64色モード
GM1	WIDTH 80,25,0,2	8色モード
GH0	WIDTH 40,25,1,2	64色モード
GH1	WIDTH 80,25,1,2	8色モード
GL2	WIDTH 40,25,0,1	64色2画面

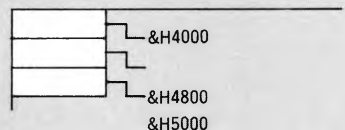
（マニュアルより抜粋）

これに新たに，GL3：WIDTH 40,25,0,1　8色モードをつけました。なおGL3の場合は32Kバイトしか使っていませんが，64K分のファイルになります。その他はすべて96Kファイルになります。

図2　X1のG-RAM構成



X1 turbo（320×200）のG-RAM構成は上記のような12枚構造です（ノーマルX1は6枚のみ)。それぞれのG-RAMの左上のアドレスは，&H4000,&H4400,&H8000,&H8400,&HC000で，7から12は裏バンクになっています。



G-RAMの左上を拡大したもので，縦に8段あって，それぞれ&H800ごとのアドレスに割りふられています。

X1turboZでも同じような構成になっていますが，4096色の場合はG-RAM12枚をまとめて2^{12}＝4096を表しています。青はG-RAMの1,2,7,8・赤は3,4,9,10・緑は5,6,11,12です。BRGそれぞれについて4ビット16階調，かたや8色は1ビット1階調（あるか，ないかだけ）です。

バッファの設定の仕方

グラフィック用バッファを確保する場合もINTEGRAL.Xを書き換えます。

1)　Z-BASICの場合

1210 MEM$(S_FF　,2)＝MKI$(&H5000)：MEM$(S_BUFF,2)=MKI$(&H6000)

1220 MEM$(S_BSIZ,2)＝MKI$(&H1000)：POKE S-IOMM,4

に差し換えてください。

2)　バンクメモリがない機種すべて

1210 MEM$(S_FF　,2)＝MKI$(&H2000)：MEM$(S-BUFF,2)=MKI$(&H3000)

1220 MEM$(S_BSIZ,2)＝MKI$(&H1000)：POKE S_IOMM,1

KAME-DOS上からCCHANGE,GLOAD,GSAVEを使う場合は上記のようにする必要があります。さらに，ノーマルX1の場合は次の1行も付け加えてください。

1225 POKE &HE139,8

2)の書き換えを行ったINTEGRAL.Xでは，CCHANGE,GLOAD,GSAVEの立ち上げ以外は行わないようにしてください。「DIR」などをすると，画面がメチャクチャになってしまいます。もとへ戻すには，書き換えを行う前のINTEGRAL.Xを起動しなおしてください。

リスト3

```
0000              ;
0000              ;
0000              ;
0000              ;
0000
C000               ORG $C000
C000
FBF6 P            #SCRNM2 EQU $FBF6
5907 P            #GRAADR EQU $5907
C800 P            #LCP    EQU $C800
CC00 P            #LCN    EQU $CC00
C000
C000 C3 16 C0      JP BEGIN
C003
C003              ;
C003 00 00        X    DW 0
C005 00 00        Y    DW 0
C007 00 00        X1   DW 0
C009 40 01        X2   DW 320
C00B 00 00        Y1   DW 0
C00D C8 00        Y2   DW 200
C00F              BRGB DS 3
C012 00           RGBP DB 0
C013 00 00        ADR  DW 0
C015 00           BIT  DB 0
C016              ;
C016
C016              BEGIN
C016 21 00 C8      LD HL,#LCP
C019 11 01 C8      LD DE,#LCP+1
C01C 01 FF 07      LD BC,$7FF
C01F AF            XOR A
C020 77            LD (HL),A
C021 ED B0         LDIR
C023 2A 0B C0      LD HL,(Y1)
C026 22 05 C0      LD (Y),HL
C029              FORY
C029 D9             EXX
C02A DD 21 00 CC    LD IX,#LCN
C02E 21 00 C8       LD HL,#LCP
C031 D9             EXX
C032 3A 05 C0       LD A,(Y)
C035 E6 01          AND 1
C037 28 08          JR Z,EXIXHL
C039 D9              EXX
C03A E5              PUSH HL
C03B DD E5           PUSH IX
C03D E1              POP HL
C03E DD E1           POP IX
C040 D9              EXX
C041              EXIXHL
C041 DD E5          PUSH IX ;LC(C,N,X)
C043 E1             POP HL
C044 5D             LD E,L
C045 54             LD D,H
C046 13             INC DE
C047 01 FF 03       LD BC,$3FF
C04A AF             XOR A
C04B 77             LD (HL),A
C04C ED B0          LDIR
C04E D9             EXX
C04F ED 5B 07 C0     LD DE,(X1)
C053 DD 19           ADD IX,DE
C055 19              ADD HL,DE
C056 D9             EXX
C057 CD 6B C0       CALL XLP
C05A 2A 05 C0       LD HL,(Y)
C05D 23             INC HL
C05E 22 05 C0       LD (Y),HL
C061 ED 5B 0D C0    LD DE,(Y2)
C065 B7             OR A
C066 ED 52          SBC HL,DE
C068 38 BF          JR C,FORY
C06A C9            RET
C06B
C06B              XLP
C06B AF            XOR A
C06C 32 0F C0      LD (BRGB),A
C06F 32 10 C0      LD (BRGB+1),A
C072 32 11 C0      LD (BRGB+2),A
C075 2A 07 C0      LD HL,(X1)
C078 22 03 C0      LD (X),HL
C07B              FORX
C07B CD 8F C0       CALL R03
C07E 2A 03 C0       LD HL,(X)
C081 23             INC HL
C082 22 03 C0       LD (X),HL
C085 ED 5B 09 C0    LD DE,(X2)
C089 B7             OR A
C08A ED 52          SBC HL,DE
C08C 38 ED          JR C,FORX
C08E C9            RET
C08F
C08F              R03
C08F CD 80 C1      CALL XYADR
C092 FD 21 0F C0   LD IY,BRGB
C096 3E 40         LD A,$40
C098 32 12 C0      LD (RGBP),A
C09B              RGBLP
C09B CD 02 C1       CALL POINT
C09E FD 7E 00       LD A,(IY+0)
C0A1 82             ADD A,D
C0A2 D9             EXX            ;
C0A3 86             ADD A,(HL)  ;LC(R,P,X)
C0A4 D9             EXX            ;
C0A5 16 00          LD D,0
C0A7 FE 78          CP 120
C0A9 38 04          JR C,SIKII
C0AB D6 78           SUB 120
C0AD 16 01           LD D,1
C0AF              SIKII
C0AF CD C7 C0       CALL LCINC
C0B2 CD 41 C1       CALL PSET
C0B5 FD 23           INC IY
C0B7 DD 23           INC IX
C0B9 D9              EXX
C0BA 23              INC HL
C0BB D9              EXX
C0BC 3A 12 C0        LD A,(RGBP)
C0BF C6 40           ADD A,$40
C0C1 32 12 C0        LD (RGBP),A
C0C4 20 D5          JR NZ,RGBLP
C0C6 C9            RET
C0C7
C0C7              LCINC
C0C7 5F            LD E,A
C0C8 CB 3F         SRL A           ;
C0CA CB 3F         SRL A           ;A/8
C0CC CB 3F         SRL A           ;
C0CE FD 77 00      LD (IY+0),A
C0D1 7B            LD A,E          ;DOWN
C0D2 E6 07         AND 7           ;
C0D4 DD 86 00      ADD A,(IX+0) ;LC(R,N,X)
C0D7 FD 86 00      ADD A,(IY+0) ;
C0DA DD 77 00      LD (IX+0),A  ;
C0DD
C0DD 2A 03 C0      LD HL,(X)
C0E0 7C            LD A,H
C0E1 B5            OR L
C0E2 28 0A         JR Z,SKIPLD
C0E4 FD 7E 00      LD A,(IY+0)  ;LEFT-DOWN
C0E7 87            ADD A,A      ;
C0E8 DD 86 FD      ADD A,(IX-3) ;LC(R,N,X-1)
C0EB DD 77 FD      LD (IX-3),A  ;
C0EE              SKIPLD
C0EE
C0EE FD 7E 00      LD A,(IY+0)  ;RIGHT-DOWN
C0F1 DD 86 03      ADD A,(IX+3) ;
C0F4 DD 77 03      LD (IX+3),A  ;LC(R,N,X+1)
C0F7
C0F7 FD 7E 00      LD A,(IY+0)  ;RIGHT
C0FA CB 27         SLA A        ;A*4
C0FC CB 27         SLA A        ;
C0FE FD 77 00      LD (IY+0),A  ;
C101 C9            RET
C102
C102              ;
C102              ;
C102              POINT ;OUT D
C102 CD 9E C1      CALL SETHLE
C105 16 00         LD D,0
C107 CD 25 C1      CALL BCTOD2
C10A 01 D0 1F       LD BC,$1FD0
C10D ED 78          IN A,(C)
C10F F6 10          OR $10
C111 ED 79          OUT (C),A
C113 CD 25 C1      CALL BCTOD2
C116 01 D0 1F       LD BC,$1FD0
C119 ED 78          IN A,(C)
C11B E6 EF          AND $EF
C11D ED 79          OUT (C),A
C11F 7A            LD A,D
C120 07            RLCA
C121 07            RLCA
C122 07            RLCA
C123 57            LD D,A
C124 C9            RET
C125
C125              BCTOD2
C125 E5            PUSH HL
C126 CD 32 C1      CALL BCTOD
C129 01 00 04      LD BC,$400
C12C 09            ADD HL,BC
C12D CD 32 C1      CALL BCTOD
C130 E1            POP HL
C131 C9            RET
C132
C132              BCTOD
C132 4D            LD C,L
C133 44            LD B,H
C134 ED 78         IN A,(C)
C136 A3            AND E
C137 28 04         JR Z,NOBIT
C139 37            SCF
C13A CB 12         RL D
C13C C9            RET
C13D              NOBIT
C13D B7            OR A
C13E CB 12         RL D
C140 C9            RET
C141
C141              ;
C141              ;
C141              PSET ;IN D
C141 CD 9E C1      CALL SETHLE
C144 CD 5D C1      CALL DTOBC2
C147 01 D0 1F       LD BC,$1FD0
C14A ED 78          IN A,(C)
C14C F6 10          OR $10
C14E ED 79          OUT (C),A
C150 CD 5D C1      CALL DTOBC2
C153 01 D0 1F       LD BC,$1FD0
C156 ED 78          IN A,(C)
C158 E6 EF          AND $EF
C15A ED 79          OUT (C),A
C15C C9            RET
C15D
C15D              DTOBC2
C15D E5            PUSH HL
C15E CD 6A C1      CALL DTOBC
C161 01 00 04      LD BC,$400
C164 09            ADD HL,BC
C165 CD 6A C1      CALL DTOBC
C168 E1            POP HL
C169 C9            RET
C16A
C16A              DTOBC
C16A 4D            LD C,L
C16B 44            LD B,H
C16C CB 1A         RR D
C16E 38 0A         JR C,BITON
C170 D5             PUSH DE
C171 7B             LD A,E
C172 2F             CPL
C173 ED 58          IN E,(C)
C175 A3             AND E
C176 ED 79          OUT (C),A
C178 D1             POP DE
C179 C9            RET
C17A              BITON
C17A ED 78         IN A,(C)
C17C B3            OR E
C17D ED 79         OUT (C),A
C17F C9            RET
C180
C180              ;
C180              ;
C180              XYADR ;IN X,Y OUT ADR,BIT
C180 2A 03 C0      LD HL,(X)
C183 ED 5B 05 C0   LD DE,(Y)
C187 AF            XOR A
C188 32 F6 FB      LD (#SCRNM2),A
C18B 06 1D         LD B,$1D
C18D ED 41         OUT (C),B
C18F CD 07 59      CALL #GRAADR
C192 06 1E         LD B,$1E
C194 ED 41         OUT (C),B
C196 22 13 C0      LD (ADR),HL
C199 7A            LD A,D
C19A 32 15 C0      LD (BIT),A
C19D C9            RET
C19E
C19E              ;
C19E              SETHLE ;OUT HL,E
C19E 3A 12 C0       LD A,(RGBP)
C1A1 2A 13 C0       LD HL,(ADR)
C1A4 84             ADD A,H
C1A5 67             LD H,A
C1A6 3A 15 C0       LD A,(BIT)
C1A9 5F             LD E,A
C1AA C9             RET
C1AB
```

X68000用画像回転プログラム

XROT0.X

読者投稿による画面回転プログラムです。比較的小さなリストでも効果てきめん。256色の画像をグルグル回します。特殊効果その他，画像処理の際に参考にしてください。なお，高速化のためエラー処理など一部省略された処理がありますので注意してください。

Watanabe Shinya
渡辺 伸也

皆さんこんばんは。拡大縮小回転というと，現在のビデオゲームを語るうえでのひとつのキーワードになっていますね。僕もアフターバーナーに感動してからパソコンでもこういうことができないかなーと思い始めました。

アフターバーナー以前にもA-JAXというものがあったようですが，僕がゲーセンに顔を出すようになったのはアフターバーナーの出る少し前あたりからなのでA-JAXの存在すらX68000への移植の話が持ち上がるまで知りませんでした（ちなみにそれ以前に最後にゲーセンに行ったときはたしかクレイジークライマーとかがあって，任天堂のゲームウォッチが流行り始めていた頃だったような）。

アフターバーナーの出た頃といえば2年以上前の話。そんな長いことかかってこのプログラムを作っていたわけではもちろんありません。このプログラムの原形は1年ほど前にすでにありましたが，「スピードを追求するあまり，画像がやたら汚い」，「エラーチェックをしないので，ちょっと使い方を間違うだけですぐにバスエラーが出る（今回投稿したこのプログラムではさらにパワーアップしていてハングアップする危険すらある）」などいろいろなアラがあって投稿作品としては失格だと考え，投稿は断念したのでした。

その少しあとにGROT.Xを目にして，「この分だと近いうちに誰かがプログラムを発表してX68000ユーザーにとって回転アルゴリズムは一般的なものになるであろう」と読んでいました。が，そうなる気配はない。アフターバーナーのX68000版が出たとき，かねがね気になっていた回転プログラムを調べてみますと（電波さんにケチをつけるつもりはありませんが）僕が開発していた過程のアルゴリズムではありませんか。

そんなわけで「こういう性格の作品が受け入れられるかどうか一度Oh!Xの読者&編集部に挑戦してみるか」と考えまして，今回の投稿へと至ったのです。

注意事項

何度もいいますがこのプログラムはスピードだけがウリで，絵はボロボロ。これで各種エラーチェック機能をつけて遅くなろうものならどーにもこーにも救いがなくなるので，エラーおよび不都合な動作に関する責任はどうしてもユーザーに負ってもらうことになります。

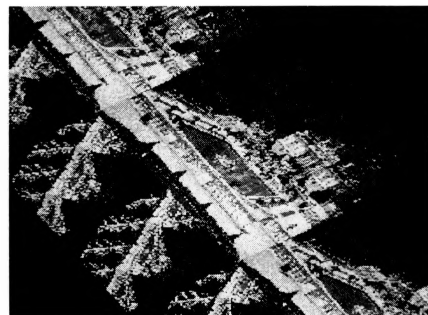
けど，ユーザー側に責任を負わせるソフトは悪いソフトだなんていえませんよね。グラフィックツールや各種言語，特にアセンブラ。よっぽどタコなソフトでなければ作品の不出来をソフトのせいにはしませんね。

マシン語なんか暴走するのが常（？）だから，誰もがテストランする前にRAMディスクの内容をセーブするし，大事なファイルの入ったディスクを実験に使ったりはしません。これはアセンブラに関しては「不都合が起こった場合の責任は自分にある」という認識が一般に広まっているからなのです。暴走したとき，真っ先に考えるのは「自分のミス」であって，「こんなはずはない！　アセンブラがバグっているのだ」なんてチラっとでも考えないはずです。

僕としてはその辺が不安の材料なわけで，「暴走するグラフィック関数」なんて皆さんはいままで見たことも聞いたこともないと思います。でもこのプログラムがそうなんです。くれぐれも注意してください。これのせいで大事なファイルが消えてしまったなんてことがないように。取り越し苦労でしょうか。

遊び方

リストはできるだけ多くの方が打ち込む気になってくれるように短くしたつもりです。まず，リストを入力します。コンパイルは，



```
CC /W XMKDAT0.C
```

です。そして実行ですが，このプログラムはカレントディレクトリに約30Kバイトのファイルを作成しますので，カレントにはその分の余裕が必要です。

プログラムを実行すると放射線が描かれ，中心が抜けていきます。放射線は360度制で3度おきに120本。抜けた中心部分は直線データとして，ファイルに吸収したのです。そしてそのファイルが出力されるXROTDAT0です。

あとはXROT0.SとTESTROT.Cを入力して，

```
CC /W /Y /w  TESTROT.C
XROT0.S
```

とすればTESTROT.Xができます。

実行時にはXROTDAT0がカレントにあるようにしてください。また，実行する前に16色または256色モードの絵をページ0にロードしておきます。絵は512×512ドットいっぱいに描き込んであるものを選んでください（65536色のデータはto256.xなどで変換してください）。

適当なデータがないときはTESTROT.Cのコメント化してある行を有効にしてみましょう。操作方法は図1ですが，まずはこちらの指示に従って操作してください。では実行です。

```
TESTROT
```

まずOPT.1キーで，縮小していきます。するとすぐに画面はページ0の大きさを越え，本来の画像の周囲に変な画像が出現します。そのあたりで操作を止めて，ページ

0の大きさを越えた画面上部の黒い部分に注目してください。拡大縮小のアルゴリズムは図2です。

この黒い部分はページ0のアドレス（$C00000）より前の領域からデータを読んでいることがわかります。しかしRAMをフル装備したマシンでない限りこの領域にメモリは存在しないわけで，普通この領域をアクセスすれば「バスエラーが発生しました」となるのですが，いまそうならないのは画像を作成しているあいだだけ，バスエラーベクタを書き換えてオリジナルのバスエラー例外処理プログラムで処理しているからなのです。このへんは実際にリストを打ち込んだ方なら察しがついていると思います。

ほとんどのユーザーのマシンでこの黒い部分の面積に比例して処理が遅くなりますが，これはバスエラーの数だけ例外処理にとんでいるためです。

絵が左右に連なっているのも含めて，このプログラムでは「絵からはみ出したかどうかチェックして回避しない」のが諸悪の根源なわけですが，その処理を入れると極端に遅くなってしまうのです。

ところで画面の下のほうに見えている「変な画像」についてですが，まずTESTROT.Xの動作を理解してください。図3です。

縮小するとサンプリング領域が広まってページ0の上をサンプリングしてバスエラーを出しますが，ページ0の下もサンプリングします。ページ0の下とはページ1，つまりいま見ている出力画面のこのページです。入力画面と出力画面が重なることによって再帰が起こり，このような画像ができるのです。

縮小ではなく移動によってページ1を見えるようにして回転とかしてみましょう。こういうのもフラクタルっていうんでしょうか？　もっと綺麗な元絵が作れれば美しい再帰画像になるんでしょうけど。

図1　TESTROT.Xのキーボード操作

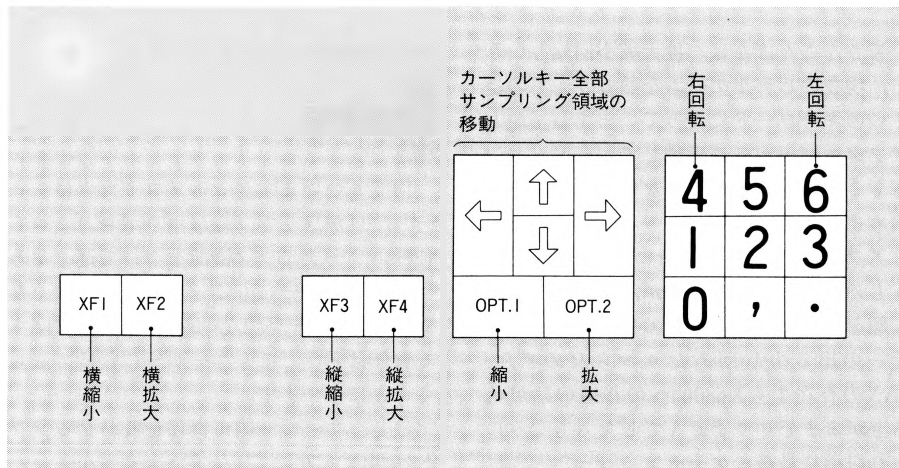


図2　拡大縮小

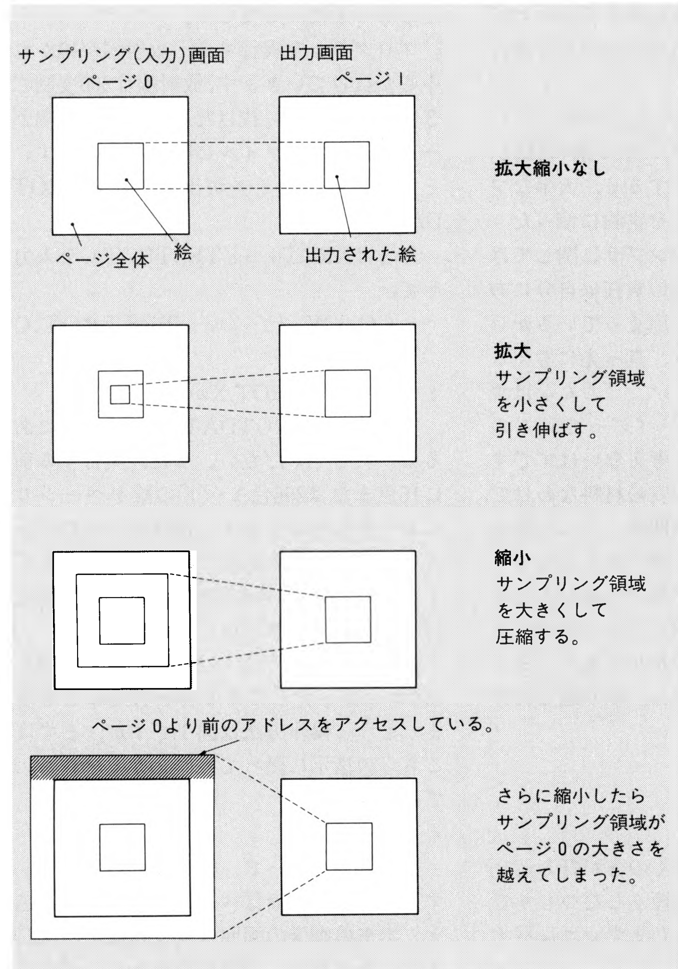
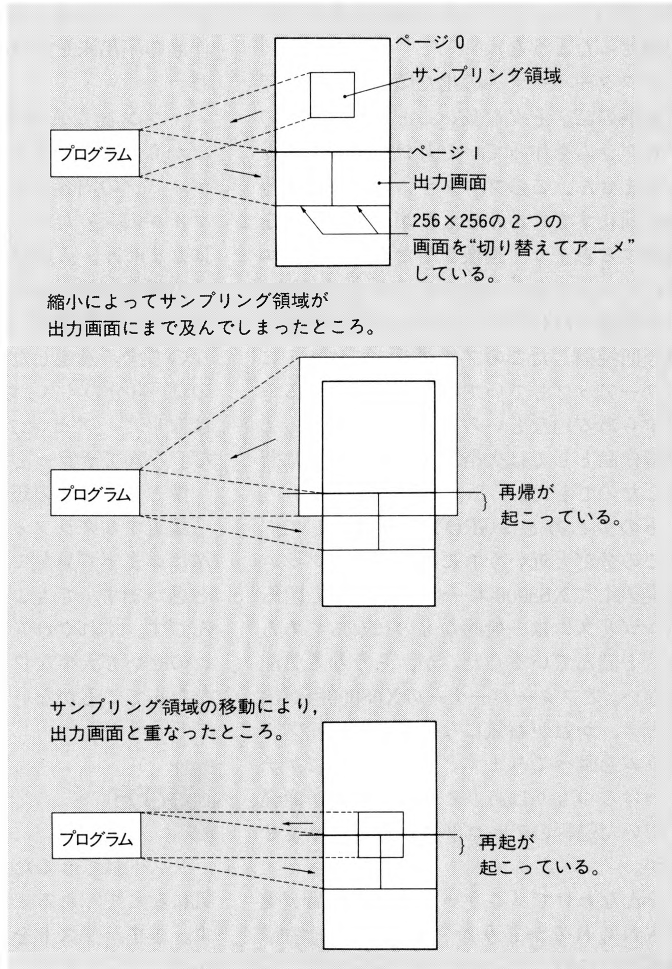


図3　TESTROT.Xの動作



ひと通りいじくって使い勝手をのみこんだらXROT0.Sのバスエラー回避部分を切ってしまいましょう。さすがにこれは危なすぎるので。描画処理中に割り込んできたプログラムがバスエラーを起こした場合にも気まずいものがあるし。あと，このバスエラー回避は読み込み側画像のエラー処理専門なので，それ以外のバスエラーだとハングアップします。

以後，その絵やページからはみ出すようなパラメータの設定をしないように気をつけましょう。

関数の説明

XROT0はいうまでもなくXC対応に作られた関数で，数Kバイトのプログラム部と，約30Kバイトのデータ部に分けられます。実際にもデータ部はXROTDAT0として別個に存在していて，プログラムが立ち上がったあとに読み込みます。本来プログラムとデータは一体化していました。これはほかのOS上（自作とか）でも動かすことを想定していたからで，Humanのコマンド（この場合はディスク関係）は使いたくなかったのです。しかしそれだとリンクが長いので今回の投稿では作り直しています。

なおXROTDAT0のあるディレクトリはXROT0.Sのデータ文で決定されるので，ここを書き換えてしまえば，どこにファイルがあろうとかまいません。

XROT0は以下の３つの関数から成り立っています。

XROT0INIT( );

ディスクからデータを読み込みます。プログラムが立ち上がったら，１回実行してください。実行しなかったら素直にバグるのみです。純正のグラフィックコマンドは「画面初期化コマンドを実行していない」ことを察知するとシカトしてくれますね。IOCSレベルからしてそういう構造になっていますが皆さんはこういうのを親切な設計だと思いますか？

IOCSといえばあのレジスタをビシバシ使うやり方はいただけません。一度作った値をメモリに待避しないでそのままレジスタに残しておいて使えるのが68000のプログラミングスタイルであり，68000の価値だと考えます。Cのようにパラメータをスタックで渡すとか，パラメータ群のセットしてあるメモリの先頭ポインタをスタックで渡すとかのほうがよいと思うのですがどうでしょう。アセンブラで組むときはまずスーパーバイザモードにして，I/Oポートの操作は自力でやる人って多いと思います。

WNDROT0(P0,P1)；

INT P0：
　入力画像のあるページ番号（０～３）
INT P1：
　画像を出力するページ番号（０～３）
　P0＝P1であってもかまわない。負数禁止。

実際にプログラムがほしがっているのは各ページ番号ではなくて，各ページの座標（0，0）のアドレスです。それをP0，P1より計算して内部に控えておきます。計算式は，

　　アドレス＝$C00000＋$80000×ページ番号

です。０～３以外の値も受け付けます。たとえば４だと$E00000，つまりテキストVRAMを指定できます。XROT0は拡大縮小回転しなければ，ただの画像転送命令として使えるので，たとえば，

　　WNDROT0(0,4)；

としてテキストVRAMに絵を置いておき，以後，

　　WNDROT0(4,0)；

とすれば，グラフィックVRAMに入力画像を置いておく必要はなくなります。

ただし，XROT0はメモリのどこを指定しても実画面512×512ドットのフォーマット（X方向のカウントが±２バイト，Y方向のカウントが±1024バイト）として扱うので，16色モードの絵を１ページ分転送してもテキストVRAMの全部を使ってしまいます。

XROT0(X1,Y1,X2,Y2,W,H,SX,SY,A);

int X1：入力画像の中心のX座標
int Y1：入力画像の中心のY座標
int X2：出力画像の中心のX座標
int Y2：出力画像の中心のY座標
int W ：出力画像の横サイズの1/2
int H ：出力画像の縦サイズの1/2
int SX：横の拡大縮小率（下位２バイトのみ有効）
int SY：縦の拡大縮小率（下位２バイトのみ有効）
int A ：回転角度　±3900000

中心の座標とはいわずと知れたその画像の縦と横の中心に位置するドットの座標ですが，中心の１ドットが存在するためには画像のサイズが縦横ともに奇数でなければなりません。

しかるに画像のサイズは1/2の状態で指定するので，サイズは必ず偶数になり真の中心ドットはなくて代わりに，候補の４ドットが存在するかたちになります。ではどうするのかというと，４ドットのうちの任意の１ドットを中心として決めてしまってよいのです。どうせ精度はガタガタなので１～３ドットの違いなど問題になりません。

ですが拡大縮小なしで回転角度が０，30，60，90，120の倍数のときはさすがに正確に動きますのでそのあたりも考慮してください。

サイズを1/2で指定させるのはこの値で計算することが多いのと，1/1の値から1/2の値を作ろうとすると，奇数だった場合に誤差が出てしまうからです（整数演算なので）。精度はガタガタだと書きましたが，なるべくそうならないようにはしているのです。

拡大縮小率の設定はややこしくて，任意の拡大縮小率の逆数を16進数の固定小数点小数として考えます。小数点は下位２バイトのあいだにとって，$00.00とします。

概念としては，

　　2倍拡大→1/2→$01.00/2→$00.80→$0080
　　1/2縮小→2/1→$01.00＊2 →$02.00→$0200

となります。

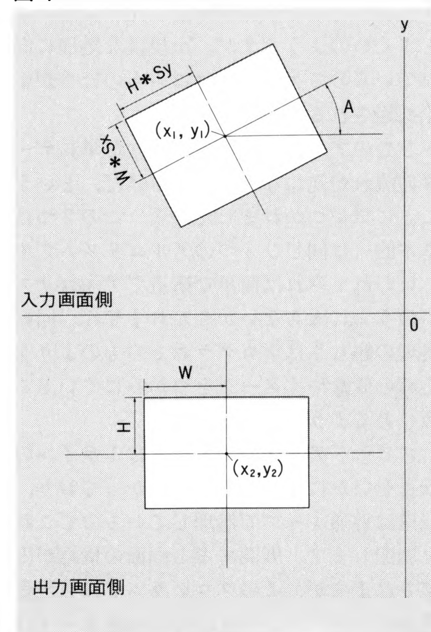
実用的な計算法は，

```
double A = 1.25；
/＊実数によるわかりやすい表記＊/
int    SX,SY;
SX = (int)( 1.0 / A ＊ 256.0 )；
```

です。

なお，いまの状態では縮小率は1/4までですが，この制限を加えているのは，XROT0.Sの18行と19行だけですので変更するのは簡単です。回転角度の制限はDIVSによる

図4



割り算の限界のことです。

回転のアルゴリズム

いろいろな回転プログラムが出まわっていますが，どれも似たような作りをしています。僕は非常に奥が深いものかと思ってさまざまな試みをしました。その結果，誰もが最初に思いつくアルゴリズムが実は一番優秀であることがわかってきました。なんか残念です。

簡単に説明すると，

「目的の画像より水平に1ライン分のデータをサンプリングし，それに回転処理を施し目的のVRAMに出力する。これを縦の大きさだけ繰り返す」

「目的の画像より回転処理を施したライン上（つまり水平垂直を含むナナメ線）に1ライン分サンプリングし，それを水平に変換して目的のVRAMに出力する。これを縦の大きさだけ繰り返す」

わけです。前者を画像回転型，後者を座標回転型と僕は呼んでいます。

この投稿は後者の座標回転型であり，描画面積が一定なため「描画速度が一定で次の絵を出すとき，前の絵をクリアすることなくそのまま上書きしてしまえる」というアニメ処理に好都合な特徴を持っています。

画像回転型は描画面積が不定なので，同じ画像データでも縮小しているほど速く処理が終わるような作り方が可能です。あるいは工夫が足りないと，拡大縮小によって処理速度が違ってくるような作りになってしまいがちです。またアニメ処理のときはいちいち前の絵をクリアする必要があります。こう書くと座標回転型に比べて悪いことずくめのようですが，小規模な処理に向いているのでゲームにはこちらのほうがよく採用されるようです。

2つのアルゴリズムは実はただ単にデータの流れが逆になっているだけだ，ということに気がつかれましたか？　この2つは基本的には同じひとつのアルゴリズムです。

いわれてみれば簡単な構造でアルゴリズムのうちに入らないかもしれません。回転処理の難しさはプログラムそのものよりも処理の概念やイメージをつかみにくいところにあるように思います。

スピードアップのポイントは1ラインの転送をいかに速くするかにかかっており，転送は普通ループで処理しているのでこれを展開します。展開すると画面の横幅が固定されますが，このプログラムでは自己書き換えによって1ライン分の転送ルーチンを「作る」ので可変長になっています。

また，この転送ルーチンの中で，

LEA d16(An),An

が使われています。これは最高速の32ビット加減算命令なのですが，イミディエイトであるd16の部分はメインメモリ上の値なのでこれを変更したくば，またもや自己書き換えです。よってこのプログラムではひとつの領域に2カ所から書き換え動作を行うことによって，ひとつの転送ルーチンを作り出しています。

XROT0は1ドット/20クロックの描画速度を持っているので，256×256ドットの画像だと秒速7コマで書き換えることができます。いろいろとサイズを変えて実験してみましょう。

あと注意が必要なのはVPAGE，HOMEなどの関数で，これらは垂直帰線期間を無視して動くので，垂直帰線期間待ちをする処理が別個に必要だということです。TESTROT.Cでは#asmでやっている部分です。Cで作ることもできますが，アセンブラで2行だと知っているとCを使う気になりません。

XROT0の拡大回転処理はウソ臭いですね。本当だったらOh!FM 3月号の「view.exp」のように拡大した四角い1ドットにも回転処理を加えなければならないところ

回転について

通常の座標系の上に回転させた座標系を想定し，その座標系上の一定領域（長方形）の中の座標を1ドットずつ順番に指定できるシステムを作る。

通常の座標系上の一定領域（長方形）の中の座標を1ドットずつ順番に指定できるシステムも作る。

そしてこの2つを同時に動かす。このときデータの流れ（受け渡し）が，

通常座標→回転座標のとき画像回転型
回転座標→通常座標のとき座標回転型

となる。

回転座標を想定しても，ドットの並びはあくまでも通常座標の方眼なので，そこに大きな無理が生ずる。それはドット画面に真の斜線が描けないのと同じである（階段になってしまう）。したがってドット構成の画面である限り，真の画像回転は不可能であり，すべて疑似的なものになる。ハードウェアによる回転でもその例にもれずチラついている。

回転処理に使う画像データはなるべくチラつきを目立たなくするためにグラデーションを多用してボカシ気味に描くのがコツである。ゲーセンに行って確かめてみよう。本当に綺麗に回転させたくば何千色も使って色の補間をする処理が必要である。

リスト1

```
===============  XMKDAT0.C  ==================
#include    "basic0.h"
#include    "BASIC.h"
#include    "graph.h"
#include    "math.h"
#include    "stdio.h"

main()
{
    int     a,b,c,sampx,sampy,ax,ay;
    short    int    x[128];
    double    DEG,SC,CO;
    FILE    *fi;
    DEG =  pi() / 180.0;
    screen( 2, 0, 1, 1 );
    home( 0, 128, 128 );
    window( 0, 0, 1023, 1023 );

    fi = fopen( "XROTDAT0", "wb" );
    for( a = 0; a < 360; a += 3 ){
        ax = (int)( 370.0 * cos( (double)a * DEG ) );
        ay = (int)( 370.0 * sin( (double)a * DEG ) );
        line( 512, 512, 512 + ax, 512 - ay, 10, 'NASI' );
        sampx =  512; sampy =  512;
        c = a / 90; c = a - c * 90;
        if( c > 45 ) c = 90 - c;
        CO = 1.;
        SC = cos( (double)c * DEG );
        for( b = 0 ; b <= 127; b++ ){
            if( point(sampx    ,sampy - 1) == 10 ){ ax =  0; ay = -1; }
            if( point(sampx + 1,sampy - 1) == 10 ){ ax =  1; ay = -1; }
            if( point(sampx + 1,sampy    ) == 10 ){ ax =  1; ay =  0; }
            if( point(sampx + 1,sampy + 1) == 10 ){ ax =  1; ay =  1; }
            if( point(sampx    ,sampy + 1) == 10 ){ ax =  0; ay =  1; }
            if( point(sampx - 1,sampy + 1) == 10 ){ ax = -1; ay =  1; }
            if( point(sampx - 1,sampy    ) == 10 ){ ax = -1; ay =  0; }
            if( point(sampx - 1,sampy - 1) == 10 ){ ax = -1; ay = -1; }

            CO = CO + SC;

            if(CO > 1.) {
                CO = CO - 1.;
              pset(sampx,sampy,0);
              sampx += ax; sampy += ay;
```

をXROT0では単にソフト的にドットを粗くしただけだったりします。

ところで，このウソ臭い回転，あのアフターバーナー（もちろん本物）がやっているのを知っていますか？　ここからは僕の憶測ですが，アフターバーナーのハードはそれまでのセガの体感シリーズであるスペハリ，エンデューロレーサー（マイナー）と同じで，スプライトには拡大縮小機能しかありませんでした。

アフターバーナーは2MバイトのRAM（と聞いた）を増設し，そこに回転パターンをこさえてパターン持ちの回転処理をするというパソコンライクな作りをしていたのです。

改造のポイント

XROT0では画像を小さく設定すれば当然処理が高速になります。が，もうひとつ高速化する方法があります。それはソースリストを書き換えることになりますが，XROT0.Sの，

197行を有効にする
200行を有効にする
201行を無効にする（コメント化する）
210行を有効にする

ことです。どうです？　なかなかうまいことやったと思いませんか？

しかし本当はプログラムを皆さんが理解して，そのうえで自力で改造していただくのが理想です。ですが本気でプログラムを解説すると何ページあっても足りないのでそれは諦めました。

その他諸々

回転というとすぐに「アサルト！」とか「ダートフォックス（メタルホークでないところがポイント。このゲーム好きなんですけど廃れるのが早いのです。しくしく。CD買いました）の移植だ！」とか聞こえてきそうですが，それは無理というものです。

処理速度の問題もありますがここを強調したいのです。XROT0は「1枚絵の回転」ですが，ナムコのシステム2は「BG画面の回転」です。BGとはX68000に搭載のあのスプライトBGのことです。ですからまるっきり違うのだということを理解してください。多くの人は回転ばかりに気を取られているようですけど。

しかし自分でプログラムを組もうとでもしてみない限りそこまで考えないのは当然

```
                }else{
                ax = ay = 0;
                }
            x[b] =  2 * ax + 1024 *  ay;
            }

        fwrite( (char *)&x, 2, 128, fi );
        printf( "%d¥n", x[127] );
        }

    for( a = 0; a <= 359; a += 3 ){
        putw( ( short int )(sin( a * DEG ) * 4096.0 ), fi );
        putw( ( short int )(cos( a * DEG ) * 4096.0 ), fi );
    }
    fclose( fi );
    screen( 2, 0, 1, 1 );
}
```

リスト2

```
===============  TESTROT.C  ==================
#include    "basic0.h"
#include    "graph.h"
#include    "doslib.h"
#include    "iocslib.h"

main()
{
    int     a,b,c,d,SSP;
    int     X1,Y1,X2,Y2,W,H,SX,SY,A;
    int     a_frag;

    SSP = B_SUPER(0);
    C_CUROFF(); A_CLR_AL();
    CRTMOD( 10 + 0x100 );

    /*******S A M P L E**************************************
    screen( 0, 2, 1, 1 );
    window(0,0,511,511);
    apage(0);
    for(a = 0; a < 16; a++ ) palet( a, rgb( 2*a, 2*a, 2*a ) );

    fill(  0,  0, 511, 511, 15 );
    fill( 10, 10, 501, 501,  0 );

    for(a = 0; a < 511 ; a += 64){
        for(b = 0 ; b < 511 ; b += 64){
            for(d = 0 ; d < 23 ; d +=4){
            c = ( a + b + c ) - ( ( a + b +  c ) / 15 ) * 15;
            fill( a+d, b+d, a+50-d, b+50-d, c );
            }
        }
    }
    box( 256-128, 256-128, 256+128, 256+128, 15 ,'NASI');
    fill( 256-40, 256-40, 256+40, 256+40, 15 );
    fill( 256-30, 200-30, 256+30, 200+30, 10 );
    ***************************************************************/

    vpage(2);
    SX = SY = 256;
    X1 = Y1 = 256;
    Y2 = 128;
     W = 128;
     H = 128;
     A = 0;
    a_frag = 1;
    XROT0_INIT();
    WNDROT0(0,1);

    while(1){
        if( BITSNS( 0x8 ) & 0x80 ) A -= 1;                 /* 4 key     */
        if( BITSNS( 0x9 ) & 0x02 ) A += 1;                 /* 6 key     */

        if( BITSNS( 0xE ) & 0x04 ) { SX += 10; SY += 10; } /* OP1 key */
        if( BITSNS( 0xE ) & 0x08 ) { SX -= 10; SY -= 10; } /* OP2 key */

        if( BITSNS( 0xA ) & 0x20 ) SX += 10;               /* XF1 key   */
        if( BITSNS( 0xA ) & 0x40 ) SX -= 10;               /* XF2 key   */

        if( BITSNS( 0xA ) & 0x80 ) SY += 10;               /* XF3 key   */
        if( BITSNS( 0xB ) & 0x01 ) SY -= 10;               /* XF4 key   */

        if( BITSNS( 0x7 ) & 0x08 ) X1 -= 5;                /* LEFT key  */
        if( BITSNS( 0x7 ) & 0x20 ) X1 += 5;                /* RIGHT key */

        if( BITSNS( 0x7 ) & 0x10 ) Y1 -= 5;                /* UP key    */
        if( BITSNS( 0x7 ) & 0x40 ) Y1 += 5;                /* DOWN key  */

        if( BITSNS( 0x0 ) & 0x02 ) break;                  /* ESC key   */

#asm

VDISP:          BTST.B     #4,$E88001                      /* 帰線待ち  */
```

です。偉ぶった文章ですがそういうつもりはありません。

なんか悲観的になりましたが「このプログラムでは無理だ」という話です。ふたたび誤解のないようお願いします。もちろん僕はBG回転に挑戦するつもりです(図5)。

BG回転機能を持ったハードを販売しているのはいまのところナムコだけと思われます。そのアーケードマシンでもロクにない機能を家庭用ゲームマシンに持ち込もうというのだから，(よくも悪くも)いかにとんでもないことをスーパーファミコンがやろうとしているかがわかると思います。コストが下げられなくて当然，発売が延びて当たり前といえますね。

*　　*　　*

XROT0ということはXROT1があるだろうと容易に想像がつくわけですね。XROT1はXROT0の20%の処理速度向上を果たしたものの，画面のサイズが128ドットまでに限られるのと仮想画面の使用を強要されるという，面白くない副作用がぞくぞくと発生したので，発表はXROT0にさせていただきました。XROT1が出ないのなら識別のために「XROT」に「0」をつけておく必要はないのですが，これは単に僕の気持ちの問題です。

おしまい

いかがでしたか？　このプログラムでX68000の限界のひとつを示したという自信があります。柴田惇氏のいい回しを借りれば「すごい」と思おうが「こんなもんか」と思おうが今後のX68000ユーザーの指針になることは確かである，というところでしょうか。あと僕としては「その筋」は死語になってほしくないので，みんなで盛り返しましょう。いろいろ偉そうなことを書いてきましたが，僕はOh!X誌上ではあまりでしゃばれないような気がします。いまさら遅いか。ではさようなら。

```
                BNE.B     VDISP

#endasm

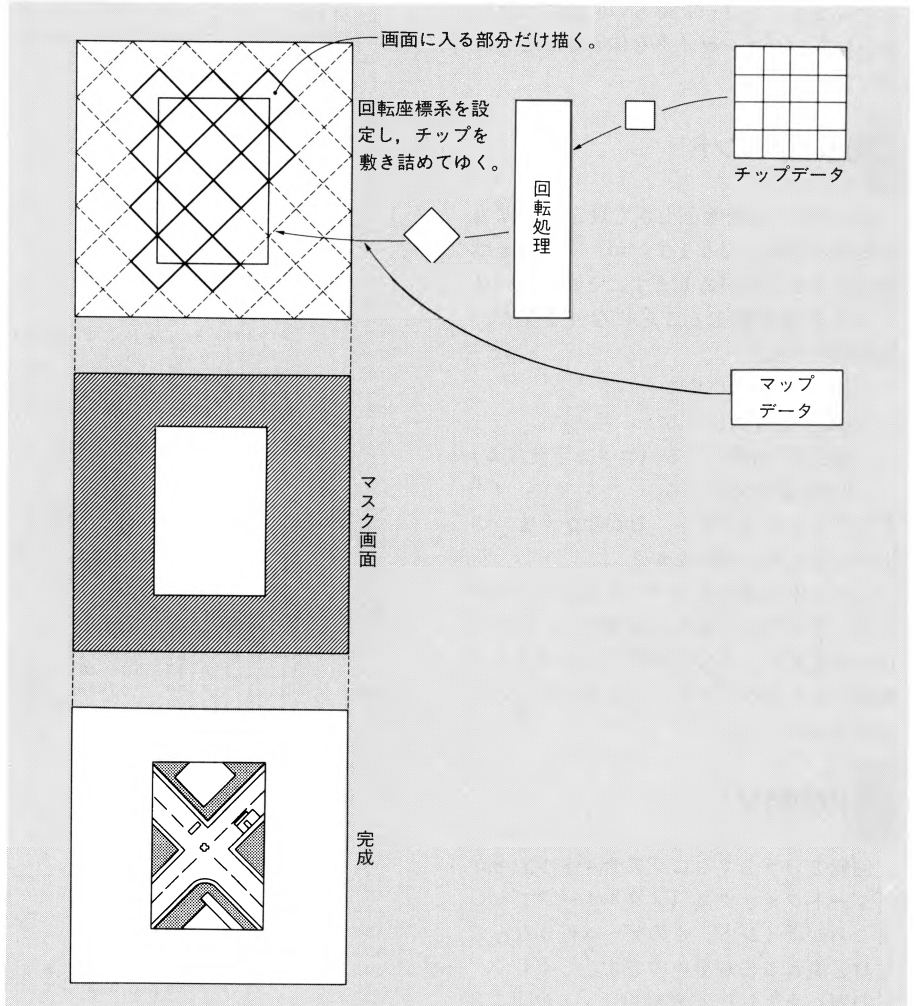
        if( a_frag == 1 ) { X2 = 128        ; home(1,256,0); }
                   else { X2 = 128 + 256; home(1,000,0); }

        a_frag *= -1;

        XROT0( X1, Y1, X2, Y2, W, H, SX, SY, A );

        }
    CRTMOD( 8 + 0x100 );
    C_CURON();VPAGE(1);
    B_SUPER( SSP );
}
```

図5　回転BGシステム



リスト3

```
================  XROT0.S  ==================
    .GLOBL  _XROT0,GO
    .GLOBL  _XROT0_INIT
    .GLOBL  _WNDROT0

    .TEXT

*************************R O T   0*********************
*S V   M O D E****************************
_XROT0:     MOVEM.L D0-D7/A0-A4,REGBUF
            MOVEQ.L #$81,D0             * _B_SUPER
            MOVEA.L #0,A1               *
            TRAP    #15                 *
            MOVE.L  D0,SSPBUF           *
            MOVE.L  USP,A0              *
            MOVE.L  A0,USPBUF           *

* E R R O R   C H E C K*****************
            AND.W   #$03FF,30(SP)       *
            AND.W   #$03FF,34(SP)       *
            CMPI.W  #128,26(SP)         * IF 128 < H THEN ERROR
            BHI.W   BAD_END             *
            MOVE.W  22(SP),D0           *
            BEQ.W   BAD_END             * IF W = 0 THEN ERROR
            CMPI.W  #128,D0             * IF 128 < W THEN ERROR
            BHI.W   BAD_END             *
            CMP.W   NOW_WIDE,D0         * W は前回の設定値と同じか
            BEQ.B   GO                  * 同じなら書き換えしない。
            MOVE.W  D0,NOW_WIDE         *

*書き換えサブルーチン************** ﾖｺ ﾉ ﾄﾞｯﾄｽｳ*2 ﾀﾞｹ ｵﾍﾟｺｰﾄﾞ ｦ ｶｲﾃ､
            LEA.L   W_LINE,A0           * BSR W_LINE ﾃﾞ ﾖﾍﾞﾙ ﾙｰﾁﾝｦ ﾂｸﾙ｡
            MOVE.W  22(SP),D0           * D0 = W
            ASL.W   #1,D0               *
            SUBQ.W  #1,D0               *
LOOP00:     MOVE.W  #$34D1,(A0)+        * #$34D1      = MOVE.W (A1),(A2)+
            MOVE.L  #$43E9_0000,(A0)+   * #$43E9_0000= LEA.L  0000(A1),A1
            DBRA.W  D0,LOOP00           *
            MOVE.W  #$4E75,(A0)         * #$4E75     = RTS

*角度取り出し*****************************
GO:         MOVE.L  36(SP),D0           * D0 = A
            MOVEQ.L #120,D1             *
            DIVS.W  D1,D0               * D0 = -120 ··· +119
```

```
         SWAP.W  D0                  *
         EXT.L   D0                  *
         ADD.L   D1,D0               * D0 =  0   ･･･  239
         DIVU.W  D1,D0               * D0 ﾉ ｱﾏﾘ=0 ･･･ 119
         SWAP.W  D0                  *
         EXT.L   D0                  *
         MOVE.L  D0,D6               *

*転送側画像  転送開始座標****************
         ASL.W   #2,D0               *
         LEA.L   XSDAT0,A0           * 三角関数データ
         MOVE.L  (A0,D0.W),D4        *
         MOVEQ.L #12,D7              * ASR ﾆﾖﾙ 1/4096 ｴﾝｻﾞﾝ ﾖｳ
         MOVE.W  D4,D5               * D5 = COS
         SWAP.W  D4                  * D4 = SIN

         MOVE.W  30(SP),D0           * D0 = SCALE X
         MULU.W  22(SP),D0           * D0 = W * SCALE X
         ASR.L   #8,D0               *
         NEG.W   D0                  *
         MOVE.W  34(SP),D1           * D1 = SCALE Y
         MULU.W  26(SP),D1           * D1 = H * SCALE Y
         ASR.L   #8,D1               *
         MOVE.W  D5,D2               * D2 = COS
         MOVE.W  D4,D3               * D3 = SIN

         MULS.W  D0,D2               * D2 = X * COS
         MULS.W  D1,D3               * D3 = Y * SIN
         SUB.L   D3,D2               * D2 = X,COS - Y,SIN
         ASR.L   D7,D2               * D2 = D2 / 4096

         MULS.W  D5,D1               * D1 = Y * COS
         MULS.W  D4,D0               * D0 = X * SIN
         ADD.L   D0,D1               * D1 = Y,COS + X,SIN
         ASR.L   D7,D1               * D1 = D1 / 4096
         MOVEQ.L #10,D7              * ASL ﾆﾖﾙ 1024 ﾊﾞｲ ｴﾝｻﾞﾝ ﾖｳ

         MOVE.W  6(SP),D4            * X1
         MOVE.L  8(SP),D5            * Y1
         ADD.W   D2,D4               *
         SUB.L   D1,D5               *
         ASL.W   #1,D4               * * 2
         ASL.L   D7,D5               * * 1024
         MOVEA.L SAMPPL,A0           *
         ADDA.W  D4,A0               *
         ADDA.L  D5,A0               * 転送開始アドレス完成

*合成側画像  描画開始座標****************
         MOVEA.L PLAYPL,A2           *
         MOVE.W  14(SP),D0           * X2
         MOVE.L  16(SP),D1           * Y2
         SUB.W   22(SP),D0           * W
         SUB.L   24(SP),D1           * H
         ASL.W   #1,D0               * * 2
         ASL.L   D7,D1               * * 1024
         ADDA.W  D0,A2               *
         ADDA.L  D1,A2               * 描画開始アドレス完成

*使用する直線データ二本の先頭アドレス****
         LEA.L   XROTDAT0,A3         *
         MOVEA.L A3,A4               *
         MOVE.W  D6,D1               * ｶｸﾄﾞ
         ASL.W   #8,D1               * 1ﾗｲﾝ ﾉ ﾃﾞｰﾀﾘｮｳ = 256 ﾊﾞｲﾄ
         ADDA.W  D1,A3               * X ﾎｳｺｳ ﾉ ﾃﾞｰﾀ ﾖﾐﾀﾞｼ ｱﾄﾞﾚｽ
         MOVE.L  D6,D1               *
         ADDI.W  #90,D1              *  90 * 3 = 270
         DIVU.W  #120,D1             * 120 * 3 = 360
         SWAP.W  D1                  *
         ASL.W   #8,D1               *
         ADDA.W  D1,A4               * Y ﾎｳｺｳ ﾉ ﾃﾞｰﾀ ﾖﾐﾀﾞｼ ｱﾄﾞﾚｽ

*「書き換えルーチン」のＬＥＡのイミディエイト値を書き換える。*
* これは横方向拡大縮小処理
         MOVE.W  30(SP),D0           * SCALE X
         MOVE.W  22(SP),D2           * W
         ASL.W   #1,D2               *
         MOVE.W  D2,D7               *
         SUBQ.W  #1,D2               *
         LEA.L   W_LINE+4,A1         *
         CLR.W   D3                  *
         TST.B   30(SP)              * IF SCALE X > 255 THEN 縮小
         BNE.B   NEXT00              *

  * 横方向拡大時の書き換え*********
LOOP01:  CLR.W   (A1)                *
         SUB.B   D0,D1               * D1 ｼｮｷｶ ﾅｼ ｶﾏﾜﾅｲ
         BCC.W   NEXT01              *
         MOVE.W  (A3,D3.W),(A1)      *
         ADDQ.B  #2,D3               * DATA READ POINTER INC
NEXT01:  ADDQ.W  #6,A1               *
         DBRA.W  D2,LOOP01           *
         BRA.B   NEXT02              *

  * 横方向縮小時の書き換え*********
NEXT00:  MOVE.W  D0,D4               *
         LSR.W   #8,D4               *
         SUBQ.W  #1,D4               *
LOOP02:  CLR.W   (A1)                *
         MOVE.W  D4,D5               *
LOOP03:  MOVE.W  (A3,D3.W),D6        *
         ADD.W   D6,(A1)             *
         ADDQ.B  #2,D3               * DATA READ POINTER INC
         DBRA.W  D5,LOOP03           *
         SUB.B   D0,D1               * D1 ｼｮｷｶ ﾅｼ ｶﾏﾜﾅｲ
         BCC.B   NEXT03              *
         MOVE.W  (A3,D3.W),D6        *
         ADD.W   D6,(A1)             *
         ADDQ.B  #2,D3               * DATA READ POINTER INC
NEXT03:  ADDQ.W  #6,A1               *
         DBRA.W  D2,LOOP02           *

* 直線データをバッファに移し、拡大縮小処理を施しておく。*
* これは縦方向拡大縮小処理
NEXT02:  MOVE.W  34(SP),D0           * SCALE Y
         ASL.W   26(SP)              * H
         SUBQ.W  #1,26(SP)           *
         LEA.L   LINEY,A3            * ﾊﾞｯﾌｧ ｾﾝﾄｳ
         MOVE.W  26(SP),D2           *
         CLR.W   D3                  *
         TST.B   34(SP)              * IF SCALE Y > 255 THEN 縮小
         BNE.B   NEXT04              *

  * 縦方向拡大時******************
LOOP04:  CLR.W   (A3)                *
         SUB.B   D0,D1               * D1 ｼｮｷｶ ﾅｼ ｶﾏﾜﾅｲ
         BCC.B   NEXT05              *
         MOVE.W  (A4,D3.W),(A3)      *
         ADDQ.B  #2,D3               * DATA READ POINTER INC
NEXT05:  ADDQ.W  #2,A3               *
         DBRA.W  D2,LOOP04           *
         BRA.B   NEXT06              *

  * 縦方向縮小時******************
NEXT04:  MOVE.W  D0,D4               *
         LSR.W   #8,D4               *
         SUBQ.W  #1,D4               *
LOOP05:  CLR.W   (A3)                *
         MOVE.W  D4,D5               *
LOOP06:  MOVE.W  (A4,D3.W),D6        *
         ADD.W   D6,(A3)             *
         ADDQ.B  #2,D3               * DATA READ POINTER INC
         DBRA.W  D5,LOOP06           *
         SUB.B   D0,D1               * D1 ｼｮｷｶ ﾅｼ ｶﾏﾜﾅｲ
         BCC.B   NEXT07              *
         MOVE.W  (A4,D3.W),D6        *
         ADD.W   D6,(A3)             *
         ADDQ.B  #2,D3               * DATA READ POINTER INC
NEXT07:  ADDQ.W  #2,A3               *
         DBRA.W  D2,LOOP05           *

* 合成側画像の改行値*********************
NEXT06:  LEA.L   LINEY,A3            *
         MOVE.W  26(SP),D0           * H
    *ASR.W       #1,D0               * 改造点
         ASL.W   #1,D7               * D7 = W
         NEG.W   D7                  *
   *ADDI.W       #2048,D7            * 改造点
         ADDI.W  #1024,D7            * 改造点

*転送開始！*******************************
         MOVE.L  $0008,BUS_ERROR     * バスエラー例外処理アドレス待避
         MOVE.L  #CANT,$0008         * オリジナル例外処理アドレス

LOOP07:  MOVEA.L A0,A1               * 描画側ライン スタートアドレス
         BSR     W_LINE              * 1ライン転送
         ADDA.W  (A3)+,A0            * 転送側座標更新
   *ADDA.W       (A3)+,A0            * 転送側座標更新（改造点）
         ADDA.W  D7,A2               * 描画側座標更新
         DBRA.W  D0,LOOP07           * 転送側ループ

         MOVE.L  BUS_ERROR,$0008     * バスエラー例外処理アドレス復帰

*ＳＶ ＭＯＤＥ ＥＮＤ*****************
BAD_END:
         MOVE.L  USPBUF,A0           *
         MOVE.L  A0,USP              *
         MOVEQ.L #$81,D0             *
         TST.L   SSPBUF              * スーパーなのに
         BMI     NEXT08              * スーパーにしようとしていた。
         MOVEA.L SSPBUF,A1           *
         TRAP    #15                 *
NEXT08:  MOVEM.L REGBUF,D0-D7/A0-A4  *
         RTS

* その場凌ぎのバスエラー例外処理     *
CANT:    LEA     8(SP),SP            *
         *LEA    $FC0000,A1          * 改造点
         CLR.W   (A2)+               *
         RTE                         *
*************************ＲＯＴ ０ ＥＮＤ**************

*************************ＸＲＯＴ０＿ＩＮＩＴ************
_XROT0_INIT:                         *
         MOVEM.L D0-D1,-(SP)         *
         MOVE.W  #0,-(SP)            *
         PEA     DATFILE             *
         .DC.W   $FF3D               * DOS _OPEN
         MOVE.W  D0,D1               *

         MOVE.L  #$7800+$1E0,-(SP)   *
         PEA     XROTDAT0            *
         MOVE.W  D1,-(SP)            *
         .DC.W   $FF3F               * DOS _READ

         MOVE.W  D1,-(SP)            *
         .DC.W   $FF3E               * DOS _CLOSE

         LEA     6+10+2(SP),SP       *
         MOVEM.L (SP)+,D0-D1         *
         RTS                         *
*************************ＸＲＯＴ０＿ＩＮＩＴ ＥＮＤ*****

*************************ＷＮＤＲＯＴ ０****************
_WNDROT0:
         MOVE.W  6(SP),D0            *
         LEA.L   $B80000,A0          *
LOOP08:  ADDA.L  #$80000,A0          *
         DBRA.W  D0,LOOP08           *
         MOVE.L  A0,SAMPPL           *
         MOVE.W  10(SP),D0           *
         LEA.L   $B80000,A0          *
LOOP09:  ADDA.L  #$80000,A0          *
         DBRA.W  D0,LOOP09           *
         MOVE.L  A0,PLAYPL           *
         RTS                         *
*********************************ＷＮＤＲＯＴ ０ ＥＮＤ*********

         .DATA
NOW_WIDE:  .DC.W  0          * 現在の横サイズ
SAMPPL:    .DC.L  $C00000    * 転送側画面の座標（０，０）のアドレス
PLAYPL:    .DC.L  $CC0000    * 描画側画面の座標（０，０）のアドレス
DATFILE:   .DC.B  'XROTDAT0',0
           .BSS
           .EVEN
USPBUF:    .DS.L  1          *
SSPBUF:    .DS.L  1          *
BUS_ERROR: .DS.L  1          *
REGBUF:    .DS.L  8+5        *
LINEY:     .DS.W  256        *
W_LINE:    .DS.W  3*256+2    * 書き換えプログラムエリア
XROTDAT0:  .DS.B  $7800      * 直線データ読込みエリア
XSDAT0:    .DS.B  $1E0       * 三角関数読込みエリア
     .END
```



X68000 CARD.FNC用カードゲーム

HEART・負けるが勝ち

Ikeya Masahiko 池谷 昌彦

読者投稿によるCARD.FNCを使用したトランプゲームです。「負けるが勝ち」という副題どおり，できるだけカードを取らないようにゲームをすすめなければいけません。なお，このプログラムの実行には1990年５月号で発表されたCARD.FNCが必要です。

私は以前より自分でカードデータを作ってゲームを作っていましたが，数度に分けてPUTするという方法では遅くてBASICでは使いものにならず弱っていました。その点，CARD.FNCはBASICでも十分実用になるので嬉しくなります。さっそく，これを使ってカードゲームを作ってみました。

ゲームの名前はHEARTです。“負けるが勝ち”と副題をつけたいと思います。このゲームは３人から６人用のトランプゲームです。４人でプレイするのがもっともバランスがとれるので，プレイヤーはコンピュータ３人と人間ひとりの構成にしました。



ゲームの内容

ゲームのルールを簡単に説明しましょう。まず，各プレイヤーは13枚ずつ手札を持ちます。親から順番に１枚ずつ手札を場にさらしていきます。このとき出せるのは親が出した台札と同じスート（記号）のカードだけです。どうしても出せない場合はなにを出してもかまいません。

カードの順位は，２，３，４，５，６，７，８，９，10，Ｊ，Ｑ，Ｋ，Ａの順に強くなっていきます（ただし違うスートのものはもっとも弱い)。１巡した時点でもっとも強いカードを出した人が次の親になります。親は取った場札を自分のところに寄せます（手札には加えない)。このなかにハートのカードがあったら，１枚につき１点のペナルティとなります。

手札がなくなった時点でペナルティの計算をし，もっとも少ない人が次のゲームの最初の親になります。

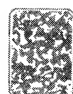




１枚もハートを取らないことをクリアといいます。クリアのときはほかの人のペナルティ分13点がもらえます。クリアが２人のときは，６点ずつで余った１点は次回に持ち越されます。

要するにハートのカードを取らないようにすればいいわけですが，ひとりでハートを13枚集めた場合だけは例外で，ほかの人から４点ずつもらえます。

だいたい感じがわかったでしょうか？



入力方法

CARD.FNCは1990年５月号で発表されたX68000用カードゲーム支援ツールです。６月号のディスクにも入っていたので，解凍して使ってください。以下にCARD.FNCをBASICに組み込むまでの手順を示します。

まず，ディスクを解凍します。６月号のオマケディスクをＢドライブに入れた場合なら，

A>LH-E B:GAMES

とすると，GAMES.LZHに入ったデータがＡドライブ上に展開されます。

ここで，BASICからMAKE.BASを実行すると自動的にCARD.FNCというファイルを作成します。できあがったCARD.FNCはBASICが入っているディレクトリに入れてください。

次に，

A>ED A:¥BASIC2¥BASIC.CNF

のように，エディタでコンフィギュレーションファイルを読み込みます。いちばん下の行に，

FUNC=CARD

と書き加え，ESC・Ｅでセーブします。これでCARD.FNCが組み込まれました。

次にBASICを起動してゲーム本体を打ち込みます。全部打ち込んだらRUNでゲームを始めてください。

＊　　＊　　＊

プログラムのシャッフル部分はCARD.FNCのサンプルである“99”からルーチンを拝借しました。このルーチンは私が使っていたものよりもよく混ざるようです。

今後もCARD.FNCを使ったトランプゲームを作っていきたいと思いますので，皆さんよろしくお願いします。

リスト１

```
10 /*
20 /*  HEART
30 /*           programed by M.I.,May22'90
40 /*
50 screen 1,1,1,1:console ,,0
60 int jj,b1,b2,b3,b4,bb,m,f=0,rd=1
70 dim int cc(51),c(3,12),pp(51),p(3,12),gg(3)
80 dim int h(12),b(3),mai(3),kei(3),kuri(4),ten(4,3)
90 dim str nam(3)={ "太郎","次郎","花子","貴方" }
100 palet(1,0)
110 /* main program
120 while f<>1
130   scrn()
140   play()
150   jd3()
160 endwhile
170 owari()
180 end
190 /* screen
200 func scrn()
210   int i
220   apage(3):vpage(15)
230   fill(0,0,511,511,3)
240   apage(2)
```

```
250   box(0,0,511,511,15):box(1,1,510,510,15)
260   line(2,144,509,144,15,&HFFFF)
270   line(160,2,160,143,15,&HFFFF)
280   line(320,2,320,143,15,&HFFFF)
290   line(2,384,509,384,15,&HFFFF)
300   line(280,145,280,383,15,&HFFFF)
310   for i=0 to 4
320     line(321,i*24+24,509,i*24+24,15,&HFFFF)
330   next
340   for i=0 to 5
350     line(i*30+360,25,i*30+360,143,15,&HFFFF)
360   next
370   symbol(26,42,"H  E  A  R  T",1,4,1,1,0)
380   symbol(24,40,"H  E  A  R  T",1,4,1,13,0)
390   symbol(376,6,"SCORE",2,1,1,15,0)
400   for i=0 to 3:symbol(324,i*24+53,nam(i),1,1,1,15,0):next
410   for i=1 to 4:symbol(i*30+341,29,str$(i),1,1,1,15,0):next
420   symbol(481,29,"合計",1,1,1,15,0)
430   symbol(348,154,"各自のハート",1,1,1,15,0)
440   for i=0 to 3
450     symbol((i mod 2)*112+286,(i ¥ 2)*104+204,nam(i),1,1,1,
15,0)
460   next
470   symbol(8,428,nam(3),1,1,1,15,0)
480   for i=0 to 3:kei(i)=0:next
490   for i=0 to 4:kuri(i)=0:next
500 endfunc
510 /* play
520 func play()
530   while rd<5
540     prep()
550     splay()
560       jd2()
570     endwhile
580 endfunc
590 /*
600 func prep()
610   int i,j,a,b,k,s
620 /* music data set
630   if rd=1 then {
640     m_init()
650     for i=1 to 8:m_alloc(i,2000):m_assign(i,i):next
660     m_trk(1,"q8@23v10o3t180l32 e")
670     m_trk(2,"q8@23v10o3t180l32 c")
680     m_trk(3,"q8@32v10o2t10l4  a")
690     m_trk(4,"q5@56v10o5t100l16aeae")
700     m_trk(5,"q7@56v10o5t 80l4  a")
710     m_trk(6,"q8@56v10o6t 50l2  c")
720     m_trk(7,"q6@1 v10o4t180c#8l12dec#dec#dec#d8e8fgefgefge
f8g8ab-gab-gab-ga4")
730     m_trk(8,"q7@19v10o4t 55b8.b16<d4>a8.b16<c>b8b8l16agf#g
a4d4b8.b16<d4>a8.b16<c4>b8b8l16ab<c>ag4")
740   }
750 /* deal
760     apage(1)
770   fill((rd-1)*30+331,25,(rd-1)*30+359,47,0)
780   fill(rd*30+331,25,rd*30+359,47,5)
790   for i=1 to 4:symbol(i*30+341,29,str$(i),1,1,1,15,0):next
800   if rd=1 then {
810     symbol(184,40,"ルールの説明は",1,1,1,15,0)
820     s=sel(176,96,1,1):if s=1 then rule()
830   }
840   randomize(val(mid$(time$,4,2)+right$(time$,2)))
850   for i=0 to 51:cc(i)=i+1:next
860   for i=0 to 3:mai(i)=0:next:m=13
870   if s=2 or rd>1 then {
880     er_upms()
890     symbol(200,24,"シャッフル",1,1,1,15,0)
900     symbol(224,56,"及び"1,1,1,15,0)
910     symbol(200,88,"カード配布",1,1,1,15,0)
920   }
930   fill(40,168,240,200,5)
940   symbol(64,176,"ちょっと待って下さい",1,1,1,15,0)
950    for i=0 to 12:h(i)=0:next
960   for i=0 to 99
970     a=int(rnd()*52):b=int(rnd()*52)
980     k=cc(a):cc(a)=cc(b):cc(b)=k
990   next
1000   fill(40,168,68,200,3)
1010   for i=0 to 51
1020     if cc(i)=1  then pp(i)=13:continue
1030     if cc(i)=14 then pp(i)=26:continue
1040     if cc(i)=27 then pp(i)=39:continue
1050     if cc(i)=40 then pp(i)=52:continue
1060     pp(i)=cc(i)-1
1070   next
1080   fill(69,168,96,200,3)
1090   for i=0 to 12
1100     c(0,i)=cc(i)    :p(0,i)=pp(i)
1110     c(1,i)=cc(i+13):p(1,i)=pp(i+13)
1120     c(2,i)=cc(i+26):p(2,i)=pp(i+26)
1130     c(3,i)=cc(i+39):p(3,i)=pp(i+39)
1140   next
1150   for i=0 to 11
1160     for j=i+1 to 12
1170       for k=0 to 3
1180         if p(k,i)>p(k,j) then {
1190           a=p(k,i):p(k,i)=p(k,j):p(k,j)=a
1200           a=c(k,i):c(k,i)=c(k,j):c(k,j)=a
1210         }
1220       next
1230     next
1240     fill(96,168,96+(i+1)*12,200,3)
1250   next
1260   er_upms()
1270   plcd()
1280   if s=1 then {
1290     click()
1300     apage(0):fill(0,0,511,511,0):apage(1)
1310   }
1320   mkba():htmai()
1330 /* play order
1340   if rd=1 then {
1350     symbol(184,24,"順番を決めます",1,1,1,15,0)
1360     symbol(176,56,"いい時にマウスを",1,1,1,15,0)
1370     symbol(208,88,"クリック",1,1,1,15,0)
1380     mouse(1)
1390     symbol(140,177,"が最初",1,1,1,1,0)
1400     repeat
1410       jj=int(rnd()*4)
1420       fill(96,177,128,192,15)
1430       symbol(96,177,nam(jj),1,1,1,1,0)
1440       msstat(i,j,a,b)
1450     until a<>0 or b<>0
1460     mouse(0)
1470     wait(50):er_upms()
1480   } else symbol(96,177,nam(jj)+"が最初",1,1,1,1,0):wait(60)
1490 endfunc
1500 /* play
1510 func splay()
1520   repeat
1530     b1=0:b2=0:b3=0:b4=0:bb=1
1540     ssplay()
1550     jd1()
1560     m=m-1
1570   until m=0
1580 endfunc
1590 /*
1600 func ssplay()
1610   while bb<5
1620     if jj>=0 and jj<=2 then {
1630       if bb=1 then {
1640         com1():bb=2:jj=jj+1
1650         if jj=3 then you():jj=0:bb=3:continue
1660       } else if bb=2 then {
1670         com(2,b2):bb=3:jj=jj+1
1680         if jj=3 then you():jj=0:bb=4:continue
1690       } else if bb=3 then {
1700         com(3,b3):bb=4:jj=jj+1
1710         if jj=3 then you():bb=5
1720       } else if bb=4 then com(4,b4):bb=5
1730     } else if jj=3 and bb=1 then you():jj=0:bb=2:continue
1740   endwhile
1750 endfunc
1760 /* com play as 1st player
1770 func com1()
1780   int i,is,hm=0,sm=0,bc=0
1790   dsban(jj)
1800   for i=0 to m-1
1810     if p(jj,i)>13 and p(jj,i)<27 then hm=hm+1
1820     if p(jj,i)<14 then sm=sm+1
1830   next
1840   while bc=0
1850     if hm>0 then {
1860       i=sm
1870       if p(jj,i)<17 then bc=1:break
1880       if p(jj,i)<18 and h(0)+h(1)+h(2)>=1 then bc=1:break
1890       if p(jj,i)<19 and h(0)+h(1)+h(2)>=2 then bc=1:break
1900       if p(jj,i)<20 and h(0)*h(1)*h(2)>0  then bc=1:break
1910       if p(jj,i)<21 and h(0)*h(1)*h(2)*h(3)>0 then bc=1:br
eak
1920       if p(jj,i)<22 and h(0)*h(1)*h(2)*h(3)*h(4)>0 then bc
=0:break
1930       if m<=5 and m=hm then bc=1:break
1940     }
1950     repeat
1960       i=int(rnd()*m)
1970     until p(jj,i)<14 or p(jj,i)>26
1980     bc=1
1990   endwhile
2000   is=i:bacd(jj,is)
2010   b1=p(jj,is):p(jj,is)=0:c(jj,is)=0:b(jj)=b1
2020   if b1>13 and b1<27 then gg(jj)=b1:h(b1-14)=1
2030   for i=0 to m-1:cc(i)=c(jj,i):pp(i)=p(jj,i):next
2040   cdleft(is)
2050   for i=0 to m-1:c(jj,i)=cc(i):p(jj,i)=pp(i):next
2060   if jj=2 then wait(15) else wait(30)
2070 endfunc
2080 /* com play as 2nd to 4th player
2090 func com(q,id)
2100   int i,is,hm=0,sm=0,ap=0,bc=0
2110   dsban(jj)
2120   for i=0 to m-1
2130     if p(jj,i)>13 and p(jj,i)<27 then hm=hm+1
2140     if p(jj,i)<14 then sm=sm+1
2150     if (p(jj,i)-1)¥13=(b1-1)¥13 then ap=ap+1
2160   next
2170   while bc=0
2180     if ap=0 then {
2190       if sm+hm>0 then is=sm+hm-1:b(jj)=0:bc=1:break else {
2200           is=m-1:b(jj)=0:bc=1:break }
2210     }
2220     if ap>0 and (b1>13 and b1<27) then {
2230       switch q
2240         case 2:is=scom2(hm,sm):break
2250         case 3:is=scom3(hm,sm):break
2260         case 4:is=scom4(hm,sm)
2270       endswitch
2280       if is>0 then bc=1
2290       if bc=0 and p(jj,sm)<22 then is=sm:bc=1:break
2300       if bc=0 and p(jj,sm)>21 then is=sm+hm-1:bc=1:break
2310     }
2320     if ap>0 and (b1<14 or b1>26) then {
2330       switch q
2340         case 2:is=sscom2():break
2350         case 3:is=sscom3():break
2360         case 4:is=sscom4()
2370       endswitch
2380       bc=1
2390     }
2400   endwhile
2410   bacd(jj,is)
2420   id=p(jj,is):p(jj,is)=0:c(jj,is)=0
2430   if ap>0 then b(jj)=id
2440   if id>13 and id<27 then gg(jj)=id:h(id-14)=1
2450   for i=0 to m-1:cc(i)=c(jj,i):pp(i)=p(jj,i):next
2460   cdleft(is)
2470   for i=0 to m-1:c(jj,i)=cc(i):p(jj,i)=pp(i):next
2480   if jj=2 then wait(15) else wait(30)
2490 endfunc
2500 /*play you
2510 func you()
```

```
2520    int i,is,x,y,l,r,ap=0,bc=0
2530    dsban(3)
2540    if b1>0 then {
2550      for i=0 to m-1
2560        if (p(3,i)-1)¥13 = (b1-1)¥13 then ap=ap+1
2570      next
2580    }
2590    while bc=0
2600      symbol(176,48,"出したいカードを",1,1,1,15,0)
2610      symbol(208,84,"クリック",1,1,1,15,0)
2620      mouse(1)
2630      msarea(49,401,502,495):setmspos(64,432)
2640      repeat
2650        msstat(x,y,l,r)
2660      until l<>0 or r<>0
2670      mspos(x,y)
2680      mouse(0):er_upms()
2690      if m>9 then is=(x-48)¥34 else is=(x-48)¥50
2700      if is>m-1 then dame():wait(40):er_upms():continue
2710      if b1>0 and ap>0 and (p(3,is)-1)¥13 <>(b1-1)¥13 then {
2720        dame():wait(40):er_upms():continue
2730      } else  bacd(3,is):b(3)=p(3,is):bc=1
2740      if b1>0 and ap=0 then b(3)=0
2750      switch bb
2760        case 1: b1=p(3,is):break
2770        case 2: b2=p(3,is):break
2780        case 3: b3=p(3,is):break
2790        case 4: b4=p(3,is)
2800      endswitch
2810      if p(3,is)>13 and p(3,is)<27 then gg(3)=p(3,is):h(gg(3
)-14)=1
2820      p(3,is)=0:c(3,is)=0
2830      for i=0 to m-1:cc(i)=c(3,i):pp(i)=p(3,i):next
2840      cdleft(is)
2850      for i=0 to m-1:c(3,i)=cc(i):p(3,i)=pp(i):next
2860      fill(2,385,509,509,0):m=m-1
2870      plcd():m=m+1
2880    endwhile
2890  endfunc
2900  /* judge1
2910  func jd1()
2920    int i,j,a
2930    dim int ba(3)
2940    for i=0 to 3:ba(i)=b(i):next
2950    for i=0 to 2
2960      for j=i+1 to 3
2970        if b(i)<b(j) then a=b(i):b(i)=b(j):b(j)=a
2980      next
2990    next
3000    for jj=0 to 3
3010      if b(0)=ba(jj) then kachi(jj):wait(40):break
3020    next
3030    a=mai(jj)
3040    for i=0 to 3
3050      if gg(i)>0 then {
3060        mai(jj)=mai(jj)+1:htcd(jj,i):htmai():gg(i)=0}
3070    next
3080    if mai(jj)>a then ha(mai(jj)-a):wait(40) else ha0():wait
(40)
3090    mkba()
3100  endfunc
3110  /* judge2
3120  func jd2()
3130    int i,j,a,b,cla
3140    m_play(6)
3150    apage(0)
3160    fill(2,145,509,383,8)
3170    box(64,208,448,352,15,&HFFFF)
3180    line(65,256,447,256,15,&HFFFF)
3190    for i=0 to 2:line(65,i*24+280,239,i*24+280,15,&HFFFF)
3200        line(313,i*24+280,375,i*24+280,15,&HFFFF)
3210    next
3220    line(120,209,120,352,15,&HFFFF)
3230    line(176,209,176,352,15,&HFFFF)
3240    line(240,209,240,352,15,&HFFFF)
3250    line(312,209,312,352,15,&HFFFF)
3260    line(376,209,376,352,15,&HFFFF)
3270    symbol(160,168,"第"+str$(rd)+"回　得点計算",1,1,2,15,0)
3280    symbol(124,217,"ハート",1,1,1,15,0)
3290    symbol(124,237,"枚　数",1,1,1,15,0)
3300    symbol(184,217,"今回の",1,1,1,15,0)
3310    symbol(184,237,"得　点",1,1,1,15,0)
3320    symbol(244,217,"前回から",1,1,1,15,0)
3330    symbol(244,237,"の繰越点",1,1,1,15,0)
3340    symbol(320,217,"修正後",1,1,1,15,0)
3350    symbol(320,237,"の得点",1,1,1,15,0)
3360    symbol(380,217,"次回への",1,1,1,15,0)
3370    symbol(388,237,"繰越点",1,1,1,15,0)
3380    for i=0 to 3
3390      symbol(76,i*24+261,nam(i),1,1,1,15,0)
3400      if mai(i)>9 then {
3410        symbol(132,i*24+261,str$(mai(i)),2,1,1,15,0)
3420      } else   symbol(148,i*24+261,str$(mai(i)),2,1,1,15,0)
3430      if mai(i)=0 then cla=cla+1
3440    next
3450    if cla=0 then {
3460      kuri(rd)=13+kuri(rd-1)
3470      for i=0 to 3:ten(rd,i)=-mai(i):next
3480    }
3490    if cla=3 then {
3500      kuri(rd)=0
3510      for i=0 to 3
3520        if mai(i)=13 then ten(rd,i)=12 else ten(rd,i)=-4
3530      next
3540    }
3550    if cla>0 and cla<3 then {
3560      kuri(rd)=13 mod cla + kuri(rd-1) mod cla
3570      for i=0 to 3
3580        if mai(i)=0 then ten(rd,i)=13¥cla else ten(rd,i)=-ma
i(i)
3590      next
3600    }
3610    symbol(264,297,str$(kuri(rd-1)),2,1,1,15,0)
3620    symbol(398,297,str$(kuri(rd)),2,1,1,15,0)
3630    for i=0 to 3
3640      if ten(rd,i)>0 then {
3650        symbol(200,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3660      } else symbol(184,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3670    next
3680    if cla=3 then {
3690      for i=0 to 3
3700        if ten(rd,i)>0 then ten(rd,i)=ten(rd,i)+kuri(rd-1):b
reak
3710      next
3720    }
3730    if cla>0 and cla<3 then {
3740      for i=0 to 3
3750        if ten(rd,i)>0 then ten(rd,i)=ten(rd,i)+kuri(rd-1)¥c
la
3760      next
3770    }
3780    for i=0 to 3
3790      kei(i)=kei(i)+ten(rd,i)
3800      if ten(rd,i)>0 then {
3810        symbol(336,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3820        symbol(rd*30+342,i*24+53,str$(ten(rd,i)),1,1,1,15,0)
3830      } else {
3840        symbol(320,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3850        symbol(rd*30+334,i*24+53,str$(ten(rd,i)),1,1,1,15,0)
3860      }
3870      fill(481,i*24+49,509,i*24+71,0)
3880      if kei(i)>0 then {
3890        symbol(493,i*24+53,str$(kei(i)),1,1,1,15,0)
3900      } else  symbol(485,i*24+53,str$(kei(i)),1,1,1,15,0)
3910    next
3920    rd=rd+1
3930    if rd<5 then {
3940      ten(rd-1,3)=ten(rd-1,3)+1
3950      for i=0 to 3:cc(i)=ten(rd-1,i):next
3960      for i=0 to 2
3970        for j=i+1 to 3
3980          if cc(i)<cc(j) then a=cc(i):cc(i)=cc(j):cc(j)=a
3990        next
4000      next
4010      for i=0 to 3
4020        if cc(0)=ten(rd-1,i) then jj=i:break
4030      next
4040      ten(rd-1,3)=ten(rd-1,3)-1
4050      symbol(176,40,str$(rd)+"回目を始めます",1,1,1,15,0)
4060      s=sel(176,96,2,2)
4070      if s=2 then f=1:rd=5
4080    } else click()
4090    apage(1):fill(0,0,511,511,0)
4100    apage(0):fill(0,144,511,511,0):er_upms():apage(1)
4110  endfunc
4120  /* judge3
4130  func jd3()
4140    int i,j,a,b
4150    if f<>1 then {
4160      vpage(9)
4170      apage(0):fill(0,0,511,511,0)
4180      for i=0 to 5
4190        box(48+i*6,80+i*6,464-i*6,432-i*6,15)
4200      next
4210      fill(79,111,433,401,2)
4220      kei(3)=kei(3)+1
4230       for i=0 to 3:cc(i)=kei(i):next
4240      for i=0 to 2
4250        for j=i+1 to 3
4260          if cc(i)<=cc(j) then a=cc(i):cc(i)=cc(j):cc(j)=a
4270        next
4280      next
4290      for i=0 to 3
4300        if cc(0)=kei(i) then jj=i:break
4310      next
4320      symbol(97,218,nam(jj)+"の優勝！",2,2,2,5,0):m_play(7)
4330      symbol(352,440,"もう１度やりますか",1,1,1,15,0)
4340      s=sel(380,465,2,2)
4350        if s=1 then {
4360          rd=1:fill(0,0,511,511,0)
4370          apage(1):fill(0,0,511,511,0)
4380          apage(2):fill(0,0,511,511,0)
4390          vpage(15)
4400        } else f=1
4410    }
4420  endfunc
4430  /* owari
4440  func owari()
4450    vpage(2):apage(1)
4460    fill(0,0,511,511,2)
4470    symbol(272,400,"お疲れ様でした",1,1,2,15,0)
4480    m_play(8)
4490  endfunc
4500  /*
4510  func scom2(hm,sm)
4520    int i,is,bc=0
4530    for i=0 to hm-1
4540      is=sm+hm-1-i
4550      if p(jj,is)<b1 then bc=1:break
4560    next
4570    if bc=0 then is=0
4580    return(is)
4590  endfunc
4600  /*
4610  func scom3(hm,sm)
4620    int i,is,bc=0
4630    for i=0 to hm-1
4640      is=sm+hm-1-i
4650      if p(jj,is)<b1 or p(jj,is)<b2 then bc=1:break
4660    next
4670    if bc=0 then is=0
4680    return(is)
4690  endfunc
4700  /*
4710  func scom4(hm,sm)
4720    int i,is,bc=0
4730    for i=0 to hm-1
4740      is=sm+hm-1-i
4750      if p(jj,is)<b1 or p(jj,is)<b2 or p(jj,is)<b3 then bc=1
:break
4760    next
4770    if bc=0 then is=0
```

▶X68000牛革ベルトなるものをもらったのだけど，仮○ライダーかなにかの変身ベルトみたいだった。
草野　貴之(19)東京都

```
4780   return(is)
4790 endfunc
4800 /*
4810 func sscom2()
4820   int i,is
4830   for i=0 to m-1
4840     is=m-1-i
4850     if (p(jj,is)-1)¥13=(b1-1)¥13 then break
4860   next
4870   return(is)
4880 endfunc
4890 /*
4900 func sscom3()
4910   int i,is,a=0
4920   if b2>13 and b2<27 then {
4930     for i=0 to m-1
4940       is=m-1-i
4950       if (p(jj,is)-1)¥13=(b1-1)¥13 and p(jj,is)<b1 then a=
1:break
4960     next
4970     if a=0 then {
4980       for i=0 to m-1
4990         is=m-1-i
5000         if (p(jj,is)-1)¥13=(b1-1)¥13 then break
5010       next
5020     }
5030   } else is=sscom2()
5040   return(is)
5050 endfunc
5060 /*
5070 func sscom4()
5080   int i,is,a=0
5090   if (b2>13 and b2<27) and (b3>13 and b3<27) then {
5100     for i=0 to m-1
5110       is=m-1-i
5120       if (p(jj,is)-1)¥13=(b1-1)¥13 and p(jj,is)<b1 then a=
1:break
5130     next
5140     if a=0 then {
5150       for i=0 to m-1
5160         is=m-1-i
5170         if (p(jj,is)-1)¥13=(b1-1)¥13 then break
5180       next
5190     }
5200   } else if (b2>13 and b2<27) and (b3<14 or b3>26) then {
5210     for i=0 to m-1
5220       is=m-1-i
5230       if (p(jj,is)-1)¥13=(b1-1)¥13 and (p(jj,is)<b1 or p(j
j,is)<b3) then a=1:break
5240     next
5250     if a=0 then {
5260       for i=0 to m-1
5270         is=m-1-i
5280         if (p(jj,is)-1)¥13=(b1-1)¥13 then break
5290       next
5300     }
5310   } else if (b2<14 or b2>26) and (b3>13 and b3<27) then {
5320     for i=0 to m-1
5330       is=m-1-i
5340       if (p(jj,is)-1)¥13=(b1-1)¥13 and (p(jj,is)<b1 or p(j
j,is)<b2) then a=1:break
5350     next
5360     if a=0 then {
5370       for i=0 to m-1
5380         is=m-1-i
5390         if (p(jj,is)-1)¥13=(b1-1)¥13 then break
5400       next
5410     }
5420   } else is=sscom2()
5430   return(is)
5440 endfunc
5450 /*
5460 func cdleft(k)
5470   int i
5480   for i=0 to m-k:cc(k+i)=cc(k+i+1):pp(k+i)=pp(k+i+1):next
5490 endfunc
5500 /*
5510 func sel(x,y,m,n)
5520   int i,j,a,b
5530   str mm,nn
5540   switch m
5550     case 1:mm="必　要":break
5560     case 2:mm="O　K"
5570   endswitch
5580   switch n
5590     case 1:nn="不　要":break
5600     case 2:nn="やめる"
5610   endswitch
5620   fill(x,y,x+56,y+24,15):fill(x+72,y,x+128,y+24,15)
5630   symbol(x+4,y+4,mm,1,1,1,1,0)
5640   symbol(x+76,y+4,nn,1,1,1,1,0)
5650   mouse(1)
5660   msarea(x+1,y+1,x+127,y+23)
5670   setmspos(x+28,y+8)
5680   repeat
5690     msstat(i,j,a,b)
5700   until a<>0 or b<>0
5710   mspos(i,j)
5720   mouse(0)
5730   if i<x+64 then {
5740    fill(x,y,x+56,y+24,1):symbol(x+4,y+4,mm,1,1,1,15,0):s=1
5750     symbol(x+4,y+4,mm,1,1,1,15,0):s=1
5760   } else {
5770     fill(x+72,y,x+128,y+24,1)
5780     symbol(x+76,y+4,nn,1,1,1,15,0):s=2
5790   }
5800   return(s):wait(40)
5810 endfunc
5820 /*
5830 func click()
5840   int i,j,a,b
5850   symbol(176,48,"よければマウスを",1,1,1,15,0)
5860   symbol(208,84,"クリック",1,1,1,15,0)
5870   mouse(1)
5880   msarea(176,48,288,96)
5890   setmspos(232,70)
5900   repeat
5910     msstat(i,j,a,b)
5920   until a<>0 or b<>0
5930   mouse(0)
5940   er_upms()
5950 endfunc
5960 /*
5970 func mkba()
5980   fill(3,145,279,383,8):fill(40,168,240,200,15)
5990   for i=0 to 3:symbol(i*56+40,344,nam(i),1,1,1,15,0):next
6000 endfunc
6010 /*
6020 func dsban(j)
6030   er_ms():symbol(108,177,nam(j)+"の番",1,1,1,1,0)
6040 endfunc
6050 /*
6060 func kachi(j)
6070   er_ms():symbol(48,177,nam(j)+"の勝ち",1,1,1,5,0)
6080 endfunc
6090 /*
6100 func ha(ht)
6110   symbol(134,177,"ハート "+str$(ht)+"枚",1,1,1,1,0):m_play
(4)
6120 endfunc
6130 /*
6140 func ha0()
6150   symbol(134,177,"ハート無し",1,1,1,1,0):m_play(5)
6160 endfunc
6170 /*
6180 func dame()
6190   er_upms()
6200   symbol(200,48,"出せません",1,1,1,5,0):m_play(3)
6210 endfunc
6220 /*
6230 func bacd(j,i)
6240   c_put(j*56+32,225,c(j,i)):m_play(1,2)
6250 endfunc
6260 /*
6270 func plcd()
6280   int i
6290   if m>9 then {
6300     for i=0 to m-1
6310       c_put(i*34+48,400,c(3,i))
6320       line(i*34+47,400,i*34+47,496,1)
6330       m_play(1,2)
6340     next
6350   } else {
6360     for i=0 to m-1
6370       c_put(i*50+48,400,c(3,i))
6380       m_play(1,2)
6390     next
6400    }
6410   symbol(16,453,str$(m),1,1,1,15,0)
6420 endfunc
6430 /*
6440 func htcd(jj,s)
6450   int a,b,h
6460   a=(jj mod 2)*112+317:b=(jj ¥ 2)*104+176
6470   if gg(s)=26 then h=14 else h=gg(s)+1
6480   c_put(a+mai(jj)*2,b,h)
6490   line(a+mai(jj)*2-1,b,a+mai(jj)*2-1,b+96,1)
6500   m_play(1,2)
6510 endfunc
6520 /*
6530 func htmai()
6540   int i
6550   for i=0 to 3
6560     fill((i mod 2)*112+294,(i ¥ 2)*104+228,(i mod 2)*112+3
10,(i ¥ 2)*104+244,0)
6570     symbol((i mod 2)*112+294,(i ¥ 2)*104+228,str$(mai(i)),
1,1,1,15,0)
6580   next
6590 endfunc
6600 /*
6610 func wait(t)
6620   int i
6630   for i=0 to t*100:next
6640 endfunc
6650 /*
6660 func er_upms()
6670   fill(161,3,319,143,0)
6680 endfunc
6690 /*
6700 func er_ms()
6710   fill(40,168,240,200,15)
6720 endfunc
6730 /*
6740 func rule()
6750   apage(0)
6760   fill(2,145,509,383,8)
6770   symbol(196,160,"ル　ー　ル",1,1,1,15,0)
6780   symbol(60,190,"1 : カードの強さは A,K,Q,J...4,3,2 の順",
1,1,1,15,0)
6790   symbol(60,208,"2 : 各自が 1 枚づつ同じ種類を出さねばなり
ません",1,1,1,15,0)
6800   symbol(120,225,"但し手持ちが無ければ、何でもかまいません
",1,1,1,15,0)
6810   symbol(60,243,"3 : 最も強い札を出した人が 4 枚全部取りま
す",1,1,1,15,0)
6820   symbol(60,261,"4 : 取った中にハートがあれば 1 枚につき -
1 点",1,1,1,15,0)
6830   symbol(60,279,"5 : 持ち札が無くなるまで繰り返します",1,1
,1,15,0)
6840   symbol(60,297,"6 : プラス点はハートを取らなかった人で分
けます",1,1,1,15,0)
6850   symbol(120,314,"1 人の時・・・ + 13 点",1,1,1,15,0)
6860   symbol(120,331,"2 人の時・・・ + 6 点、残りは繰り越し",
1,1,1,15,0)
6870   symbol(60,349,"7 : 1 人でハート 13 枚取れば +12 点、他の
人は -4 点",1,1,1,15,0)
6880   apage(1)
6890 endfunc
```

▶最近，どんどん，どんどん体力がなくなってゆく。どうしよ？　金子　聡(18)新潟県



トランジェントコマンドを作る

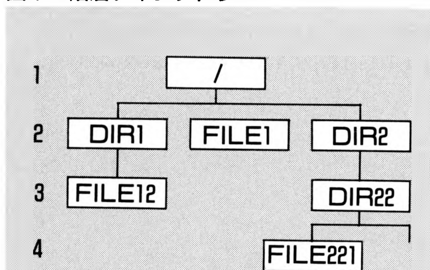
亀田 雅彦 Kameda Masahiko

KAME-DOSをもっとDOSらしく使うための方法として，KAME-DOSの外部コマンドを作成してみましょう。ディスク管理のほか，さまざまなプログラムがコマンドとして使用できます。こういったものがBASICで記述できるのです。

先月号でノーマルX1にも対応して，いよいよ本格的になってきました。もし，X1ユーザーでまだKAME-DOSを入手していない方は，ぜひバックナンバーなどから入力するようにしてください。

6，7月号のプログラムだけではなかなかその威力を発揮しないKAME-DOSも，今月から紹介していく外部コマンドを活用すればその世界が広がります。特にノーマルX1ユーザーには，ディスク関係の命令がturboBASICに匹敵するようになるので，お楽しみに。また，外部コマンドのノウハウが蓄積していくとユーザー自身の手でKAME-DOSワールドを広げていくことができるようになります（もちろん最初の公約どおり，BASICで）。

図1 階層ディレクトリ



このように，ディレクトリの下にまた下位ディレクトリを作ってファイルの整理をしやすくする構造を階層ディレクトリといいます。1段目のことをルートディレクトリと呼んで，ここから「見える」のは，2段目だけで3段目以降は見えません。また，DIR1にいるときはルートディレクトリやDIR2とは関係なくなるので，同じファイル名を使っても上書きされません。本文中の「フルパス」というのは，ファイル名をルートディレクトリから全部表示したものです。FILE221をフルパスにすると，

/DIR2/DIR22/FILE221

になります。ルートディレクトリは「/」（スラッシュ）です。カレントディレクトリというのは，現在自分のいるディレクトリで「/とかDIR1とかDIR22」となります。

とりあえず，今月から何回かに分けて，普通のDOSにあるような命令を外部コマンドとして発表しながら，その動作と作り方を説明しましょう。基本的に外部コマンドも内部コマンドも（COMMAND.X1内に用意されてるもの），作りは同じなのでCOMMAND.X1の理解の助けにもなると思います。

それでは，今月は「MD.X1」「RD.X1」そして，特集と関連して「GLOAD.X1」「GSAVE.X1」を発表してみましょう。

外部コマンドワールド

リスト3が「MD.X1」です。turboBASICでいうところのMKDIRにあたります。下位ディレクトリをカレントディレクトリの下に作ることですが，難しいところなので階層ディレクトリ全般について簡単に図1で説明しておきます。また，階層ディレクトリとは切っても切れない関係にあるCD（ディレクトリの変更）命令については，内部コマンドなので6月号に解説されています（でも，6月号ではちょっと手抜き解説が多かったと反省することしきりです）。

反省ばかりしていても進歩がないので，さっそく使い方に入りましょう。

命令：MD（MKDIR）
書式：MD 新規ディレクトリネーム
プログラム：リスト3

まずリスト3を打ち込んでください。使用BASICは，いま自分の持っているINTEGRAL.Xが動いているものならなんでも大丈夫です（CZ-8FB01ver1.0, turboBASIC,Z-BASIC)。用途別に自分でBASICシステムを構築してください。ただしCZ-8FB01では日本語入力ができないので，リストの一番最後にDATA文としてまとめられてるメッセージは，注釈行のほうを生かして日本語のほうは打ち込まなくて結構です。たとえばリストで，

1650 LABEL “d1”: DATA エラーが発生しました!!

1660 'LABEL “d1”: DATA Error !!

という2行は，

1650 '

1660 LABEL “d1”: DATA Error !!

というようにします。これが2行ずつ組になっているので，それぞれについて変更してください。日本語入力ができて，しかも使用中に日本語表示ができる（をしたい）場合には（ディスプレイの関係で表示できないこともある)，そのまま入力してください。以後，外部コマンドの入力形式はだいたい同じようなかたちになります。

使い方：

入力したら，カレントドライブかパスの通っているドライブにセーブしてください。INTEGRAL Xのコマンドライン（[X:/]の状態）から，セーブしたときのファイル名（この場合は「MD.X1」か「MKDIR.X1」）をタイプしてリターンキーを押してください。「MD」か「MKDIR」だけで，拡張子はいりません。

6月号でも書いたことですが，拡張子が「.X1」のBASICファイルはKAME-DOSの外部コマンドとして認識されます。見かけ上は，内部コマンドの実行となんら変わりありません。また，コマンドラインからパラメータとして与えられる新規ディレクトリネームの書式については，囲みを参照してください。

パスが通ってなかったり，ファイル名をタイプミスしたときはエラーになります。エラーが起きずに，しばらくすると外部コマンドがロードされて起動します。指定に

間違いがなければ，下位ディレクトリを作成して，パスに従って「COMMAND.X1」をロードしなおしてコマンドラインに復帰します（CP/Mでいうリブート）。ここで，外部コマンド実行の際の注意点を挙げておきましょう。

1)　外部コマンドのファイル名は，内部コマンドのコマンド名にあたるものなのでわかりやすくすること（片仮名などにするとあとで苦労します）。拡張子は「.X1」にすること。

2)　外部コマンド実行中にブレイクして実行を強制的に中止したときは，必ず「COMMAND.X1」を実行するところから始めてください。外部コマンドをブレイクしてそのままRUNすると，変数がクリアされるので最悪の場合暴走します。これは入力したプログラムをデバッグしているときも同じことで，エラーが出て止まったら，入力ミスを訂正していったんセーブして，「COMMAND.X1」をRUNしてそのコマンドラインから外部コマンドを実行するようにしてください（図2）。

3)　「COMMAND.X1」は必ずパスの通っているドライブにセーブしておいてください。リブートするときにパスの順に従って「COMMAND.X1」を探すので，みつからないと「リブートできません」というメッセージが出て実行が止まります。コマンドラインからの実行のときとは違って，カレントドライブでもパスが通ってないと探しにいきません。

4)　外部コマンドからリブートした時点で，下位ディレクトリにいてもすべてルートディレクトリに戻されます。たとえば，[A:/TEST/] から「MD」を実行して戻ってくると，[A:/] になっているということです。これが外部コマンドと内部コマンドが見かけ上異なる唯一の点です。

上記のうち，特に2)が大切なので必ず守ってください。このほかにも外部コマンド実行中にさまざまなエラーが発生する可能性がありますが，その場合はエラーメッセージを出力し，実行を中止して「COMMAND.X1」へ復帰しようとします。エラーメッセージは個々の外部コマンドが持っているものなので，統一されていません。

以上のことは，外部コマンド全般についていえることなので，これからも覚えておいてください。

命令：RD（RMDIR）

書式：RD　消去するディレクトリネーム

プログラム：リスト4

MDの逆で既存のディレクトリを消します。使用BASICも，その日本語部分の入力の仕方もMDと同じです。

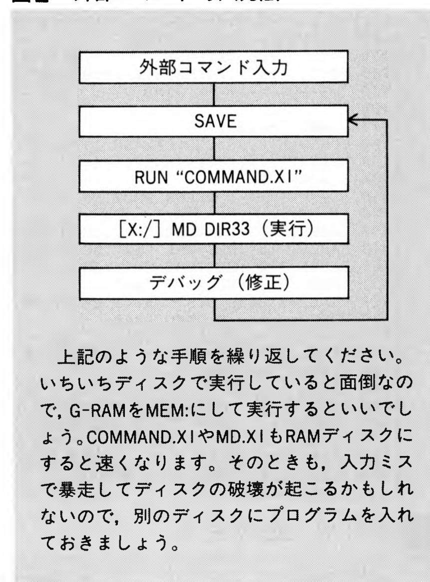
使い方：

セーブする際の注意や，実行の仕方についてもMDのところを見てください。機能的にはturboBASICのものと同じです。ディレクトリ内にファイルが残っているときは，消去できません。ファイルをDELコマンドですべて消してから実行してください。

以上，２つの外部コマンドを紹介しましたが，これによって作成されたディレクトリなどは完全にBASICとの互換性があるので，ファイルのやりとりも自由にできます。

でも，CZ-8FB01には階層ディレクトリ機能がないので，KAME-DOSで作った下位ディレクトリにはBASIC側からはアクセスできません。なお，BASICのみならずMS-DOSフォーマット（2D，2HD）とも互換性があるので，MS-DOSディスクにディレクトリを作成しようとすれば自動的にプログラム側で判断して，MS-DOSフォーマ

図2　外部コマンドの入力法



上記のような手順を繰り返してください。いちいちディスクで実行していると面倒なので，G-RAMをMEM:にして実行するといいでしょう。COMMAND.X1やMD.X1もRAMディスクにすると速くなります。そのときも，入力ミスで暴走してディスクの破壊が起こるかもしれないので，別のディスクにプログラムを入れておきましょう。

ファイルネーム

ディレクトリの名前も，基本的にそのディスクフォーマットのファイル名と同じです。ファイル名の方は各マニュアルをみてもらうなり，６月号にも少し解説しておきました。turboBASICの場合ディレクトリの拡張子は「.DIR」になるので，それにあわせておきました。フルパスで指定もできますし，カレントドライブからの指定もできます。図１のDIR22の下にDIR33を作りたいのなら，ルートから「MD /DIR2/DIR22/DIR33」か，DIR22から「MD DIR33」です。消したい場合は，MDをRDに変えてください。

リスト1　GLOAD.X1

```
1000 'GLOAD.X1  Ver 1.0        By Kameda
1010 '
1020 DEFINT a-z:IF PEEK(&HD07F) THEN KLIST 0
1030 CONSOLE 0,25
1040 DEFUSR1=m_opens:DEFUSR2=m_preop
1050 '
1060 iomm=PEEK(v_iomm):badr$=MEM$(v_badr,2):ff$=MEM$(v_ff,2)
1070 MEM$(s_ff,2)=MKI$(&H2000)
1080 bsiz!=&HC0*&H100:MEM$(v_bsiz,2)=MKI$(bsiz!):POKE v_iomm,1
1090 POKE v_dn,PEEK(s_dn):IF fe$(1)="" THEN "!2"
1100 POKE &HE137,4:POKE v_mac,PEEK(s_mac4+PEEK(v_dn))
1110 IF PEEK(&HD07F)=0 THEN dirg=PEEK(&HE139):POKE &HE139,8
1120 '----------( MAIN ROUTINE )------------
1130 '
1140 GOSUB 1380
1150 k=PEEK(v_stop):IF k=3 THEN "!3" ELSE IF k<>0 THEN "!"
1160 CLS:f$=MEM$(v_fnam+13,3):k=PEEK(&HD07F)
1170 '
1180 IF f$="GL0" OR f$="gl0" THEN IF k THEN WIDTH 40,25,0,1 ELSE WIDTH 40
1190 IF f$="GL1" OR f$="gl1" THEN IF k THEN WIDTH 80,25,0,1 ELSE WIDTH 80
1200 IF f$="GM0" OR f$="gm0" THEN IF k THEN WIDTH 40,25,0,2 ELSE WIDTH 40
1210 IF f$="GM1" OR f$="gm1" THEN IF k THEN WIDTH 80,25,0,2 ELSE WIDTH 80
1220 IF f$="GH0" OR f$="gh0" THEN IF k THEN WIDTH 40,25,1,2 ELSE WIDTH 40
1230 IF f$="GH1" OR f$="gh1" THEN IF k THEN WIDTH 80,25,1,2 ELSE WIDTH 80
1240 IF f$="GL2" OR f$="gl2" THEN IF k THEN WIDTH 40,25,0,1 ELSE WIDTH 40
1250 IF f$="GL3" OR f$="gl3" THEN IF k THEN WIDTH 40,25,0,1:OPTION SCREEN 4 ELSE
     WIDTH 40
1260 INIT:IF k THEN POKE v_wfd0,PEEK(&HF8D6)
1270 GOSUB 1380:IF PEEK(v_stop) THEN "!"
1280 '
1290 POKE v_iomm,1:CALL m_devi:IF PEEK(v_stop) THEN "!"
1300 IF PEEK(v_iofg)=0 GOTO 1320
1310 POKE v_iomm,2:CALL m_devi:IF PEEK(v_stop) THEN "!"
1320 '
1330 GOSUB "ending"
1340 POKE &HE137,6:POKE v_iomm,iomm:MEM$(s_ff,2)=ff$:MEM$(v_badr,2)=badr$
1350 CONSOLE 0,24:IF PEEK(&HD07F) THEN KLIST 1 ELSE POKE &HE139,dirg
1360 proces=proces-1:CHAIN proces$(proces)
1370 '----------(    OPEN     )-----------
1380 '
1390 MEM$(v_badr,2)=MKI$(&H3000)
```

```
1400 POKE v_ddrv+1,7,1:POKE v_iofg,0:POKE s_escp,0:fe$=fe$(1)
1410 POKE v_od,1:d$=USR2(fe$):fe$=RIGHT$(fe$,PEEK(v_yen))
1420 IF PEEK(v_stop) RETURN
1430 POKE v_sbdr,1:POKE v_op,0:d$=USR1(fe$)
1440 MEM$(v_badr,2)=MKI$(&H4000):RETURN
1450 '----------(     END       )----------
1460 '
1470 LABEL "ending"
1480 CONSOLE 0,25:CLS:CFLASH 1:PRINT "PUSH SPACE":CFLASH 0
1490 REPEAT:A$=INKEY$:UNTIL A$=" "
1500 CLS:RETURN
1510 '----------( ERROR ROUTINE )----------
1520 '
1530 LABEL "!4":RESTORE "m3":GOTO 1570
1540 LABEL "!3":RESTORE "m2":GOTO 1570
1550 LABEL "!2":RESTORE "m1":GOTO 1570
1560 LABEL "!" :RESTORE "m0"
1570 READ m$:BEEP:CLS:CREV 1:PRINT m$;:CREV 0:PRINT
1580 POKE v_stop,0:GOTO 1340
1590 '----------( DATA AREA     )----------
1600 LABEL "m0":DATA エラーが発生しました！！
1610 'LABEL "m0":DATA Error !!
1620 LABEL "m1":DATA ファイル・ネームを指定してください
1630 'LABEL "m1":DATA Need FILE-NAME
1640 LABEL "m2":DATA ファイルがみつかりません
1650 'LABEL "m2":DATA FILE Not Found
1660 LABEL "m3":DATA リブートできません
1670 'LABEL "m3":DATA Not REBOOT
```

リスト2　GSAVE.X1

```
1000 'GSAVE.X1  Ver 1.0       By Kameda
1010 '
1020 DEFINT a-z:INIT:IF PEEK(&HD07F) THEN KLIST 0
1030 CONSOLE 0,25
1040 DEFUSR1=m_opens:DEFUSR2=m_preop
1050 '
1060 CLS:LOCATE 10,10:PRINT "SAVE GRAM= [1] 96K"
1070     LOCATE 10,12:PRINT "           [2] 64K":COLOR 5
1080     LOCATE 10,14:PRINT "[space]=GRAPHIC ON OFF":COLOR 7
1090     LOCATE 10,16:PRINT "   PUSH [1] or [2]";
1100 k=0:REPEAT:a$=INKEY$(1):sx=VAL(a$)
1110 IF a$=" " THEN IF k=0 THEN k=1:SCREEN ELSE k=0:PALET
1120 UNTIL 1<=sx AND sx<=2:PRINT sx
1130 '
1140 iomm=PEEK(v_iomm):badr$=MEM$(v_badr,2):ff$=MEM$(v_ff,2)
1150 MEM$(s_ff,2)=MKI$(&H1800):MEM$(v_badr,2)=MKI$(&H3000)
1160 IF sx=1 THEN bsiz!=&HC0*&H100 ELSE bsiz!=&H60*&H100
1170 MEM$(v_bsiz,2)=MKI$(bsiz!):POKE v_iomm,1
1180 POKE v_dn,PEEK(s_dn):IF fe$(1)="" THEN "!2"
1190 POKE &HE137,4:POKE v_mac,PEEK(s_mac4+PEEK(v_dn))
1200 IF PEEK(&HD07F)=0 THEN dirg=PEEK(&HE139):POKE &HE139,8
1210 '----------( MAIN ROUTINE )------------
1220 '
1230 GOSUB 1520:IF PEEK(v_stop)<>0 THEN "'"
1240 MEM$(v_badr,2)=MKI$(&H4000)
1250 '
1260 i=1:k=15:IF m=2 THEN k=1 ELSE IF m=4 THEN k=2
1270 IF sx=2 THEN 1330
1280 POKE v_iomm,1:POKE v_od,2:POKE v_iofg,2:POKE v_edr,0:CALL m_devi
1290 IF PEEK(v_stop) THEN "!"
1300 POKE v_iomm,2:POKE v_od,2:POKE v_iofg,2:POKE v_edr,k:CALL m_devi
1310 IF PEEK(v_stop) THEN "!"
1320 fx$=MKI$(&H2000):fm$=MKI$(&H8000)+MKI$(1):GOTO 1390
1330 '
1340 POKE v_iomm,1:POKE v_od,2:POKE v_iofg,2:POKE v_edr,0:CALL m_devi
1350 IF PEEK(v_stop) THEN "!" ELSE MEM$(v_badr,2)=MKI$(&HA000)
1360 POKE v_od,2:POKE v_iofg,2:POKE v_edr,k:CALL m_devi
1370 IF PEEK(v_stop) THEN "!"
1380 fx$=MKI$(&HC000):fm$=fx$+MKI$(0)
1390 '
1400 z=1:f$=fx$+MKI$(0):IF m=2 OR m=4 THEN z=0:f$=fm$
1410 POKE v_zoku+2,z:MEM$(v_fszl,4)=f$
1420 POKE v_od,2:MEM$(v_badr,2)=MKI$(&H3000):CALL m_saved:CLS
1430 IF PEEK(v_stop) THEN "!"
1440 '
1450 CFLASH 1:PRINT "PUSH ANY KEY":CFLASH 0
1460 REPEAT:A$=INKEY$:UNTIL A$<>"":CLS:POKE &HE137,6
1470 IF PEEK(&HD07F)=0 THEN POKE &HE139,16
1480 POKE v_iomm,iomm:MEM$(s_ff,2)=ff$:MEM$(v_badr,2)=badr$
1490 CONSOLE 0,24:IF PEEK(&HD07F) THEN KLIST 1 ELSE POKE &HE139,dirg
1500 proces=proces-1:CHAIN proces$(proces)
1510 '----------(    OPEN      )-----------
1520 '
1530 POKE v_ddrv+1,1,7:POKE v_iofg,0:POKE s_escp,0:fe$=fe$(1)
1540 POKE v_od,2:d$=USR2(fe$):fe$=RIGHT$(fe$,PEEK(v_yen))
1550 m=PEEK(v_mac):IF PEEK(v_stop) RETURN
1560 POKE v_sbdr,1:POKE v_op,3:d$=USR1(fe$):RETURN
1570 '----------( ERROR ROUTINE )----------
1580 '
1590 LABEL "!3":RESTORE "m2":GOTO 1620
1600 LABEL "!2":RESTORE "m1":GOTO 1620
1610 LABEL "!" :RESTORE "m0"
1620 READ m$:BEEP:CLS:CREV 1:PRINT m$;:CREV 0:PRINT
1630 POKE v_stop,0:GOTO 1440
1640 '----------( DATA AREA     )----------
1650 LABEL "m0":DATA エラーが発生しました！！
1660 'LABEL "m0":DATA Error !!
1670 LABEL "m1":DATA ファイル・ネームを指定してください
1680 'LABEL "m1":DATA Need FILE-NAME
1690 LABEL "m2":DATA リブートできません
1700 'LABEL "m2":DATA Not REBOOT
```

ットのディレクトリを作ります。ユーザーはフォーマットの違いを意識する必要はありません。これを使えば「X68000のディスクをX1turboZで編集する」といったことも可能です。

グラフィックローダ/セーバ

グラフィック特集にあわせて，画面のロード/セーブを行うプログラムをKAME-DOS上で開発しました。特集のほうのプログラムはturboZオンリーですが，このローダとセーバはX1シリーズ全機種で使用可能です。Z-BASIC以外の標準BASICにはこのような命令がなかったので，X1間での画像のやりとりも多少便利になると思います。詳しい説明は特集記事に譲るので，ここではその紹介だけしておきましょう。

命令：GLOAD

書式：GLOAD　ファイルネーム

プログラム：特集を参照

使い方はほかの外部コマンドと同じです。特集のプログラムから子プロセス的に呼び出されるので，通常の外部コマンドとは少し異なります。

命令：GSAVE

書式：GSAVE　ファイルネーム

プログラム：特集を参照

GLOADと同じ。

解説！プログラミング

今月は短くて，しかもBASICプログラムなので難しいことはありません。ですから「外部コマンドの作成作法について」を中心に展開してみましょう。

第1部　起動

まず，コマンドラインからコマンド名が打ち込まれました。COMMAND.X1はそれが内部コマンドではないと判断して，ドライブにコマンドと同じファイル名を探しにいきます。なければエラーで，あれば拡張子が「.X1」かどうか（外部コマンドかどうか）を見て処理を振り分けます（図3）。

外部コマンドならCHAINして，そうじゃなければRUNします。ここが重要で，CHAINによってCOMMAND.X1で定義された変数がそのまま引き継がれます。外部コマンド側では必要に応じてその変数を

図3　外部コマンドの動作



使うことになります。だから，実行中にプログラムを止めて再実行することができないのです。これは必要な変数を何度も定義しないようにして，外部コマンド側の負担を軽くするためです。

それならば，ここでいう必要な変数とはなんでしょう？　内部・外部に関わらずコマンドを実行するときには，KAME-DOS共通のD000ₕ番地以降に常駐しているマシン語プログラムをアクセスします（7月号のアセンブルリストのこと)。マシン語オンリーで開発しているのならアドレスはラベルに固定できますが，BASICによる開発だとアドレスを変数に定義して，ラベルとして使う必要があります。いわばこれらはグローバル変数で，コマンド内でのみ使われるのがローカル変数というところです。なお，COMMAND.X1をリブートすると一度すべての変数をクリアするので，使用変数がたまりすぎることはありません。

第2部　実行

外部コマンドはその利用目的によって相当異なった作りになるので，一言ではいいきれないものがあります。ただ，大きく分けると次の3つになります。

1)　ファイルを扱うコマンド
2)　ディスクを扱うコマンド
3)　それ以外のコマンド

1)は，主に内部コマンドに採用されているものでCOPYやDIRなどになります（もちろんCOPYと同じことをする外部コマンドを作ることも可能です)。特徴として，ファイルをアクセスする前には必ずそのファイルをOPENし，書き込んだあとにはCLOSEするということです。そのためOPEN/CLOSEルーチンを呼び出す必要があります。また，実際にファイルの中身をアクセスするルーチンも使われるでしょう。開発する場合は，一番面倒なコマンドになります。

2)はFORMATやDISKCOPYのことです（今月は発表できませんが，そのうちに発表したいと思います)。これらはディレクトリとかFATとか，ランダムアクセスの部分がいらないので比較的簡単に開発できます。マシン語ルーチンも低級な（ハードを直接アクセスする）ルーチンを使うのでわかりやすくなります。

3)は，特にKAME-DOS上で開発する必要はないようなプログラムです。ご存じのとおりKAME-DOSはディスクアクセスルーチンの集合体です。そのうえでディスクを使わないようなプログラムを動かしても，マシン語の常駐部分だけメモリの無駄になります。開発環境も整備されていないので，このようなプログラムは発表しないつもりです。

今月のMD,RD,GLOAD,GSAVEは1)にあたります。MD,RDの解説で，外部コマンドの雰囲気をつかんでください。

第3部　リブート

COMMAND.X1へリブートするときには，それ専用のマシン語ルーチンが用意されています。注意点は前に書いてあるとおりです。GLOAD,GSAVEに関してはこのルーチンを使わずに単なるCHAIN命令で済ませていますが，普通はリブートルーチンを使います。最終的にはCHAINを使うので，COMMAND.X1に戻ったときの結果は同じです。

これは別にCOMMAND.X1へのリブートだけじゃなくて，「PROCES$」という変数で管理されているひとつ上の親プロセスへ戻るために使います。つまり，COMMAND.X1というのは親プロセスというかたちになっています。

MDとRD

どちらのプログラムでもまず，DEFUSRを定義しています。これらは先に解説したマシン語ルーチンのアドレスです。変数の頭に「M,V,S」がついているのはCOMMAND.X1からの持ち越し変数です。次に，いわゆる「OPEN」と「ディレクトリ名が指定してあるかどうか」のチェックをしま

MS-DOSのディレクトリ

MS-DOSフォーマットのディレクトリ管理方法は，X1のそれとちょっと違っています。下位ディレクトリの先頭には，「.」「..」というファイル名が2つ記録されています。これはMS-DOSでは「カレントディレクトリ」と「親ディレクトリ」を表していて，そのクラスタ番号も記録してあります。実際にこれらを使って管理しているかどうかはわかりませんが，X1にはこれに相当するものがありません。そこで，KAME-DOSでは上記のようなファイル名が出てきたら無視を決め込みます。親ディレクトリのクラスタ番号は，内部ワークエリアに保存しておくようにしました。

リスト3　MD.X1

```
1000 'MD (MKDIR)    ver 1.0     By M.Kameda
1010 '
1020 DEFUSR0=m_opens:DEFUSR1=m_preop:DEFUSR2=&HEE80:DEFUSR3=m_tranr
1030 '
1040 POKE v_dn,PEEK(s_dn):POKE v_mac,PEEK(s_mac4+PEEK(s_dn))
1050 POKE v_od,1:GOSUB 1260:IF PEEK(v_stop) THEN "erre"
1060 IF fe$(1)="" OR fe$(1)="/" THEN "errx"
1070 ON PEEK(v_mac) GOSUB 1370,1350,1370,1390
1080 GOSUB 1450
1090 POKE v_edw,k:POKE v_zoku+1,i1:MEM$(v_msbt,2)=MKI$(i0)
1100 MEM$(v_bf,2)=MKI$(buff):POKE v_frwf,1:CALL m_crsrw
1110 POKE v_frwf,0:IF PEEK(v_stop) THEN "erre"
1120 CALL m_saved :IF PEEK(v_stop) THEN "erre"
1130 d$=USR3(proces$(proces-1)):IF PEEK(v_stop) THEN "errb"
1140 k=PEEK(v_dn):IF k<4 THEN DEVICE STR$(k)+":"+RIGHT$(STR$(3-PEEK(v_mac)),1)
1150 proces=proces-1:CHAIN MEM$(v_p256+&H81,PEEK(v_p256+&H80))
1160 '_______ ERROR
1170 '
1180 LABEL "erre":RESTORE "d1":GOTO 1200
1190 LABEL "errx":RESTORE "d2"
1200 READ m$:PRINT:CREV 1:PRINT m$;:CREV 0:PRINT:POKE v_stop,0:GOTO 1130
1210 LABEL "errb":RESTORE "reb"
1220 READ m$:PRINT:CREV 1:PRINT m$;:CREV 0:PRINT:d$=INKEY$(1)
1230 POKE v_stop,0:GOTO 1130
1240 '_______ SUB
1250 '
1260 LABEL "open"
1270 d$=USR1(fe$(1)):IF PEEK(v_stop) THEN RETURN
1280 fe$(1)=RIGHT$(fe$(1),PEEK(v_yen)):IF fe$(1)="" OR fe$(1)="/" RETURN
1290 k=PEEK(v_mac):d=INSTR(fe$(1),".")
```

```
1300 IF (k=1 OR k=3) AND d=0 THEN fe$(1)=fe$(1)+".DIR"
1310 POKE v_sbdr,2:POKE v_op,3:d$=USR0(fe$(1))
1320 MEM$(v_fszl,4)=CHR$(0,0,0,0):MEM$(v_fnam1+46+22,5)=CHR$(0,0,0,0,0)
1330 RETURN
1340 '
1350 LABEL "ms"
1360 k=1:i0=1024:i1=&H10:d=0   :RETURN
1370 LABEL "x1"
1380 k=1:i0=256 :i1=&HC0:d=&HFF:RETURN
1390 LABEL "m2"
1400 k=2:i0=1024:i1=&H10:d=0   :RETURN
1410 '
1420 LABEL "poke"
1430 d$=USR2(MKI$(p)+MKI$(1)+CHR$(j)):p=p+1:RETURN
1440 '------- DATA
1450 LABEL "mem"
1460 MEM$(&HEE80,16)=HEXCHR$("EB 5E 23 56 23 4E 23 46 23 CD 93 EE 13 0B 78 B1")
1470 MEM$(&HEE90,16)=HEXCHR$("20 F7 C9 7E C3 27 E0 00 00 00 00 00 00 00 00 00")
1480 MEM$(&HEE95,2)=MKI$(m_lddea)
1490 d$=USR2(MKI$(buff)+MKI$(i0)+CHR$(d))
1500 i=PEEK(v_mac):IF i=1 OR i=3 RETURN
1510 '_______ for MS-DOS
1520 RESTORE 1600
1530 p=buff   :FOR i=0 TO 11:READ j:GOSUB "poke":NEXT
1540 p=buff+32:FOR i=0 TO 11:READ j:GOSUB "poke":NEXT
1550 p=buff+26:j=PEEK(v_crs):GOSUB "poke":j=PEEK(v_crs+1):GOSUB "poke"
1560 p=buff+58:i=PEEK(v_csdir+PEEK(v_dn)):IF i=0 THEN 1590
1570 h=v_csdir+26+8*PEEK(v_dn)+2*(i-1)
1580 j=PEEK(h):GOSUB "poke":j=PEEK(h+1):GOSUB "poke":RETURN
1590 j=0:GOSUB "poke":j=0:GOSUB "poke":RETURN
1600 '
1610 DATA 46,32,32,32,32,32,32,32,32,32,32,16
1620 DATA 46,46,32,32,32,32,32,32,32,32,32,16
1630 '_______ MESSAGE
1640 '
1650 LABEL "d1":DATA エラーが発生しました！！
1660 'LABEL "d1":DATA Error !!
1670 LABEL "d2":DATA ファイル名を指定して実行してください
1680 'LABEL "d2":DATA What file-name?
1690 LABEL "reb":DATA リブートできません
1700 'LABEL "reb":DATA reboot error
```

リスト4　RD.X1

```
1000 'RD (RMDIR)    ver 1.0     By M.Kameda
1010 '
1020 DEFUSR0=m_opens:DEFUSR1=m_preop:DEFUSR2=m_tranr
1030 '
1040 POKE v_dn,PEEK(s_dn):POKE v_mac,PEEK(s_mac4+PEEK(s_dn))
1050 IF fe$(1)="" THEN "errx"
1060 POKE v_od,1:GOSUB 1270:k=PEEK(v_stop):POKE v_stop,0
1070 IF k=0 THEN "erry" ELSE IF k<>3 THEN "erre"
1080 GOSUB 1320:IF fe$(1)="" THEN "errx"
1090 k=PEEK(v_stop):IF k=3 THEN "errz" ELSE IF k THEN "erre"
1100 CALL m_dlfat:CALL m_dldir:CALL m_clos2
1110 '
1120 d$=USR2(proces$(proces-1)):IF PEEK(v_stop) THEN "errb"
1130 k=PEEK(v_dn):IF k<4 THEN DEVICE STR$(k)+":"+RIGHT$(STR$(3-PEEK(v_mac)) 1)
1140 proces=proces-1:CHAIN MEM$(v_p256+&H81,PEEK(v_p256+&H80))
1150 '_______ ERROR
1160 '
1170 LABEL "erre":RESTORE "d1":GOTO 1210
1180 LABEL "errx":RESTORE "d2":GOTO 1210
1190 LABEL "erry":RESTORE "d3":GOTO 1210
1200 LABEL "errz":RESTORE "d4"
1210 READ m$:PRINT:CREV 1:PRINT m$;:CREV 0:PRINT:POKE v_stop,0:GOTO 1120
1220 LABEL "errb":RESTORE "reb"
1230 READ m$:PRINT:CREV 1:PRINT m$;:CREV 0:PRINT:d$=INKEY$(1)
1240 POKE v_stop,0:GOTO 1120
1250 '_______ SUB
1260 '
1270 LABEL "open"
1280 d$=USR1(fe$(1)):IF PEEK(v_stop) THEN RETURN
1290 fe$=RIGHT$(fe$(1),PEEK(v_yen))
1300 POKE v_sbdr,0:POKE v_op,1:d$=USR0(fe$):RETURN
1310 '
1320 LABEL "dopen"
1330 i=INSTR(fe$(1),"/"):IF i THEN 1380
1340 d$=USR1(fe$(1)):IF PEEK(v_stop) THEN RETURN
1350 k=PEEK(v_dn):j=PEEK(v_csdir+k):POKE v_fnam1+46+43,j
1360 i=v_csdir+26+k*8+(j-1)*2:POKE v_fnam1+46+44,PEEK(i),PEEK(i+1)
1370 POKE v_sbdr,2:POKE v_op,2:d$=USR0(fe$(1)):RETURN
1380 '
1390 fe$="":WHILE i
1400 fe$=fe$+LEFT$(fe$(1),i):fe$(1)=RIGHT$(fe$(1),LEN(fe$(1))-i)
1410 i=INSTR(fe$(1),"/")
1420 WEND:IF fe$(1)="" THEN RETURN
1430 d$=USR1(fe$):IF PEEK(v_stop) THEN RETURN
1440 GOTO 1370
1450 '_______ MESSAGE
1460 '
1470 LABEL "d1":DATA エラーが発生しました！！
1480 'LABEL "d1":DATA Error !!
1490 LABEL "d2":DATA ファイル名を指定して実行してください
1500 'LABEL "d2":DATA What file-name?
1510 LABEL "d3":DATA ディレクトリにファイルがあります
1520 'LABEL "d3":DATA File exists
1530 LABEL "d4":DATA 指定されたディレクトリがありません
1540 'LABEL "d4":DATA No directory
1550 LABEL "reb":DATA リブートできません
1560 'LABEL "reb":DATA reboot error
```

す。エラーは1カ所にまとめて同一の処理がなされます。

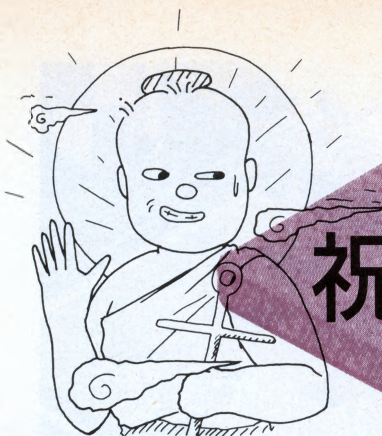
MDではMD独自のマシン語プログラムを持っています。これはディレクトリ領域初期化の高速化のためです。そして，このように短いマシン語プログラムを使う場合は，EE00$_H$番地からの256バイトを使うことになります。ここは汎用ワークエリアなので保存はしておけませんが，一時的に置いておくことはできます（ほかの外部プログラムでもこうしていくつもりです）。

その後ろにはMS-DOS用の特別初期化ルーチンが続いています。実際の書き込みは「mcrsrw」ルーチンをコールすることで行われます。そして，エラーがなければ「msaved」ルーチンでいま書き込んだディレクトリをCLOSEします。これらのルーチンは，ただコールしただけじゃ正常には動きません。その前後で盛んにPOKEしているように，あらかじめ値を設定しておかなければならないのです。POKEアドレスの意味は7月号のアセンブルリストを見ればわかるでしょう。

そして最後はリブートルーチンです。USR3命令からCHAIN命令までがそうで，これはRDでも同じです。USR3で親プロセスを引数にして，その結果はVP256＋&H81からに格納されています。この内容はフルパスファイルネームです。ディスクが入れ替えられている可能性も考慮して，DEVICE命令も実行しています。

RDでも基本的な作りは同じですが，OPENとCLOSEの部分が違っています。OPENが2つに分かれているのは，「ディレクトリ自体のOPEN」と「そのディレクトリ内にファイルがあるかどうかを調べるOPEN」があるからです。CLOSEの場合は，ファイルを消すのとわけが違って，3回に分けたコールが必要になります。そのほかには，初期化する必要もないのでMDのようなマシン語ルーチンはありません。

これで外部コマンドの概要はわかってもらえたと思います。まだ作るにはいたらないかもしれませんが，わかるところを改造してみるのもいいでしょう。来月はもっと突っ込んだ説明をして，なにか新しいコマンドを発表しながら，実際にコマンドが作れるようになるくらいまではやりたいと思っています。



Oh!X 通巻100号記念特別企画

対戦ポピュラス
祝一平VS西川善司

実況・解説　浦川博之

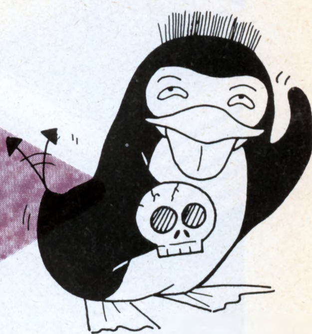


illustration: J.Yamada

編集室で対戦ポピュラスなんかやられちゃ面白くって大迷惑。なのに西川君が祝氏に挑戦状をFAXで送っちゃうんだから，さあ大変。100号記念なのに，もっと実のある企画はないのかぁ～，といいつつOh!X史上最大の決戦の火蓋は切って落とされた。

5月上旬のある日。そもそも編集室にはX68000が2台並んでいるのが悪い。これで対戦ポピュラスをやるなったって無理というもの。かくして今日もスタッフの対戦が行われるわけです。なかでもズバ抜けて強いのが西川善司。270面を制覇し，対戦は負けたことがないとか。

善「まぁ，ぼくにかなう人はいないかな」

編「いや，祝さんがAmigaで始めて，いま420面だからわかりませんよ。ね，祝さん」

善「フフフ，負けませんよ，祝さん」

祝「(ニヤッと笑って中指を立てる)」

すでにこの会話以来，2人の対決は必然だったのです。

＊

5/28　18:30決闘当日。

善「来た，祝さん」

祝「……いたな，青二才めが」

善「ひょっとしてあのFAX，怒ってる？」

祝「叩き潰してくれる」

おおっと，出会い頭にこのエキサイトぶり。おや，観客の中に丹明彦さんの姿が。丹さん，丹さんは善司くんを負かす寸前まで追いこんだそうですね。

丹「ええ，向こうが何もできなくなるところまでいったんですが，いつの間にか逆転されてしまいました。はは」

お，祝氏が自分のマウスと専用マット（なぜか航空機力学の本）を持って現れた。

善「道具まで気にしちゃって，もう」

と言いつつ，善司くんもマット代わりのフロッピーケースを取りに戻っている。

さて，今回の対戦のマップはレビューを書いた中野修一氏が作った特製だということです。中野さん，ちょっとすいません，どんなマップか教えていただけます？

中「ええ，いろいろあります」

へぇ，たとえば？

中「(ニヤリと笑って) 結構スゴイです」

うーん，この人も意味不明な気合いが入ってるな。さて……。

中「どのマップにします？」

祝「じゃあ，この砂漠のにしよう。異存はないな？」

善「どのマップでも同じですよ。へへ」

おおおおとギャラリーが沸く。

ここでちょっとマップの説明を。地形は完全に対称で，お互い人口は1人ずつでスタート。奇跡はすべて起こせます。ひとつ変わっているのは，沼が“底無し”に設定されている点。普通は1人沼に落ちるとそのマスは平地に戻るんだけど，このマップではいつまでたっても人が落ち続けるというわけ。沼を作られたら最優先で直さないとマズいわけですな。

20:53　さあ，ゲームスタート。

祝「あれ，人はどこにいるんだ？」

中「(ニヤリと笑って) え，いるじゃないですか」

どどーん。中央にそれらしく島を作っておきながら，人はマップの隅に，しかも岩に囲まれて細々とテントを立てていた。これじゃあ思うように家を増やせない。確かにスゴいマップだあ。

善「あーっ，ちくしょう。でいでい」

あああ，むりやりテントを城にする気か。中野氏が「あっあっあっ」と心配そうにも嬉しそうな悲鳴をあげる。

家を作らせてもらえない悪魔の民。ふらふらとさまよっているうちに……。

YOU LOST
SOCRE 2570

あーっ，なんと開始後1分で西川善司の連勝記録ストーップ！　あまりの情けなさにギャラリーは開いた口がふさがらない。

丹「体力もないのに砂漠を歩かせたりするから……」

祝「うっはっは，口ほどにもない」

善「しまった。気にしすぎたぁ」

中「だからぁ，もうちょっと気合い入れませんか？」

狙いどおりの展開に嬉しそうな中野氏。

気合いを入れ直して，20:55再開。今度は2人とも慎重に人が増えるのを待っているようです。

祝「死ぬなよ死ぬなよ……よぉーし！」

城は小さい家に比べて人の増えるペースが速いが，収容人員が多いため人があふれるのには時間がかかる。ということは，「城をときどき壊して人を追い出す」というのが常用テクニックになるわけです。2人は次々と城を作っては追い出し平地を開拓しています。左上のマップを見ていると，平たい大陸がじわじわ中央に向かって伸びていくのがブキミ。

ズゴゴゴゴゴ。おぉーっと，祝氏の領土に地震。最初にしかけたのは善司くんだぁ。

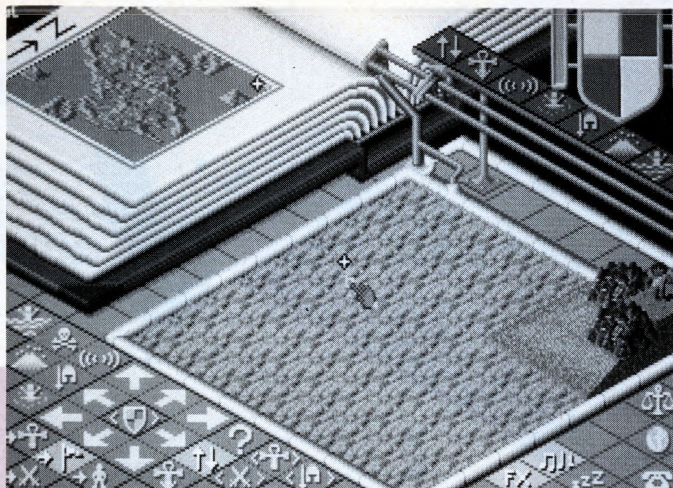
祝「地震なんか効かないもん」

さっさと修復してしまう祝氏。マウスのクリックにムダがない。さすが420面はダテじゃないぞ。

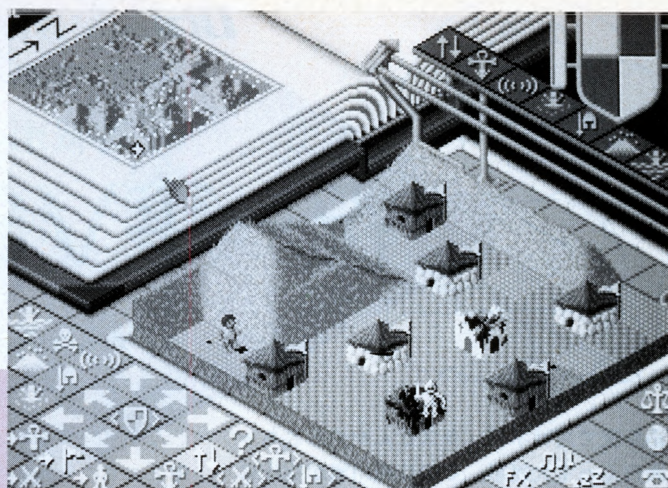
Oh!X通巻100号に寄せて

ども。ポピュラスやりたさにパソコンを買ってしまった浦川です。いま100面ちょっとですけどね。え？　機種ですか？　よ，弱ったな，なんでもいいじゃないですか。

まぁ，自宅でじっくりとコンピュータをイジメるのもいいですが，人間相手となるとまた格別。うひひ。Oh!X編集部にはたくさんX68000がありまして（当たり前だ），その中には2台RS-232Cでつながれたヤツもあります。これを見ると，さもX68000が「ぼくたち対戦ポピュラスのためにここにいまーす」と言っているような気がして，ついついそこら辺の人に「やろーやろー」と声をかけてしまう。かくして編集者がドアを開けて入ってくるなり，締め切り間際のライターがマウスをカチカチやっている姿に頭をかかえるという日々が続くのでした。おっと，これのどこが100号記念の祝辞なんだろう。というわけで合掌。



これが1回戦のマップの初期状態。左右対称の大陸と丹念に岩が配置されている。ところどころに沼も置いてあるという。ちなみにリーダーはいない。



戦況はかなり煮詰ってきた。かつての海はどこへやら……。下が祝一平，上が西川善司。手作りの山攻撃が城や町を破壊している。もう泥沼。

ゴボシ　あれ？

ゴボシ　何の音だ？

ゴボシ　ぬ，沼だ！

だはははと無責任に笑うギャラリー（なぜか驚嘆より笑いが先に立ってしまう）。

祝「やりおったな」

善「いえ，なんにもしてませんよぉ」

祝「と，いうことは……」

ギャラリー全員の視線が中野氏に集まる。

善「まさか，最初っからあるんじゃぁ」

中「まぁ見てのお楽しみ」

やはり中央の島に沼があった。新大陸に家を建てようと勇んで出かけた民は，この沼に沈んでいたわけ。中野氏恐るべし。

祝「うーん，砂漠の沼は発見しづらい」

善「中野さん，あらかじめ設定しとくなんてすごいイジワル（**ズゴゴゴゴ**）」

しっかりスキを見て地震を起こす善司くん。

丹「砂漠で地震は効きますよ」

へぇ，なんで？

丹「ほら，外をうろちょろしてる間に体力がなくなっちゃうから」

善司くんはマナが貯まるたびに地震をしかけます。対照的に祝氏はマナを貯めながらひたすら領土を拡大。

21:35　ふたりの領土がそろそろ接してきました。善司くんが一番敵地に近い家を探して，画面の端にくるように設定している。

カチカチカチカチ……

みるみる相手の土地が盛り上がる。ワッハッハと無責任に笑うギャラリー。でた。これが善司くんの得意技，手作りの山だ。

祝氏は地震で素早く取り壊す。しかし修復し終えたころにはマップのほかの場所でずんずんと巨大なピラミッドが立っている。

祝「むぅ（**シュィィィン**）」

おーっと，怒った祝氏が火山をお見舞いだー！　しかも二段重ね！

*

22:00　あれから1時間。手作りの山と火山が乱れ飛んで，かつての平地はどこへやら。家の数を見ると善司くんのほうが押し気味ではあるけど，人口ゲージを見るとまだまだ互角。人数が多いので次第に処理速度も落ちてきた。しかもハングアップ防止のため2400ボーでやっているのでなおさら。マウスの反応が悪くてときどきヘンなところがぼこっと盛り上がったりしています。

22:30　開始から1時間半たって戦いはやや膠着状態に。そろそろ休憩にしません？

祝「向こうが泣いて頼むんだったら休んでやってもいいよ」

善「もう，祝さんったら強情なんだから。素直に休みたいと言えばいいのに」

祝「なに，そんなに休みたいの？」

善「まさか。祝さんが泣いて頼むんだったらべつですけど」

次第に善司くんがじわじわと平地を獲得している。やはり手作りの山の対応に追われ続けている祝氏の不利は否めない。ところで祝さんが手作りの山はほとんどしかけないのは，なにか信条があるのだろうか？

祝「おい，休んでやってもいいよ」

善「いいですよ，べつに」

祝「……休んでやってもいいんだよ」

善「だからいいってば（**ブォン**）」

ああっと，騎士が誕生。対戦ではよほど有利でないとできない行為だ。散在する祝氏の家を焼いてまわる騎士。さらに手作りの山攻撃が襲いかかる。これらを全部修復しながら挽回をはかるのは祝氏といえども至難の技だ。

祝「むっ。くそっ。くそっ」

脂汗をにじませながら力をこめてクリックを続ける祝氏。反応が鈍いんだから，そんなに力をこめたって……。

祝「うるさい。やってるほうの身にもなってみろ」

ついにいっぱいだった人口ゲージも減少を始めた。騎士が次から次へと送りこまれ，あっちこっちで山が立つ。祝氏側の家は端のほうに散在するばかり。

そして23:07。

祝「……うむ。今日のところは負けにしといてあげよう」

ついに祝氏敗北宣言！　西川善司のTKO勝ちで決着！

祝さん，敗因は？

祝「若さに負けた」

2時間20分の長丁場ですからね。

祝「それから，あの沼は発見しづらいからキライ。そもそもマップを作ったあのコミッショナーが悪い」

勝った善司くんは？

善「そうねえ，へへへ。まぁ，丹さんのほうが強かったかな。なんちて。ぽっくん」

祝「この借りは必ず返すぞ」

善「いつでも来なさい。はっはっは」

*

その4日後。

「ちわーす」編集室に入っていくと，さっそく再戦している2人の姿があった。

善「祝さんが泣いて頼むからさあ」

祝「この前のは練習。今度が本番」

2人とも好きにしてちょうだい。

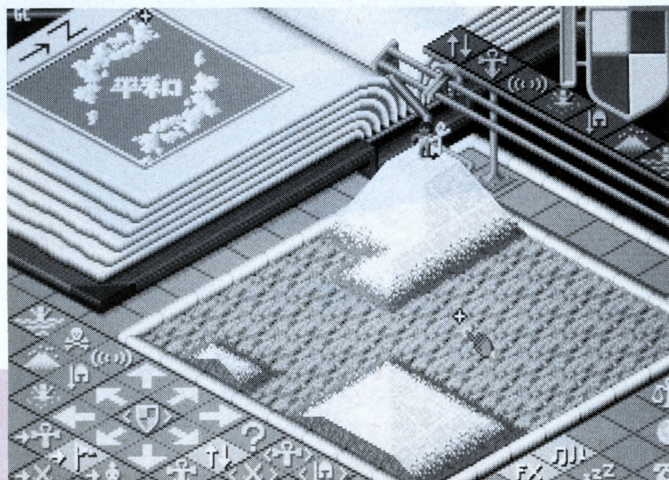
今度はもっと素直なマップで対戦。雪原に点対称に日本が2つ配置され，沖縄に1人だけ人間がいるという設定です。

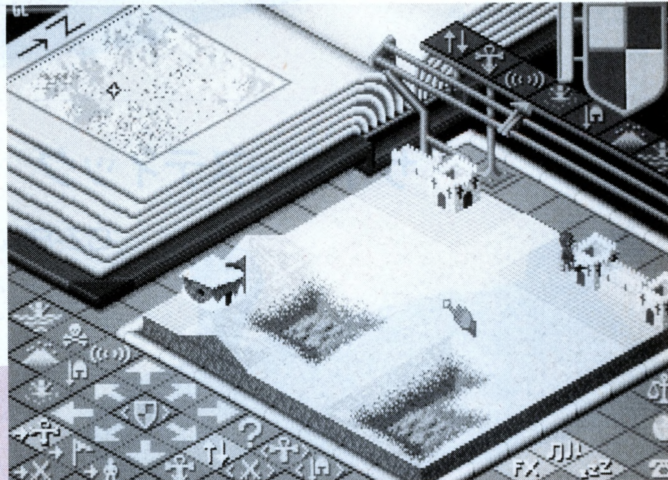
おや，祝氏が家をくずして，一番低い平地で展開するのに対して，善司くんは一段

▶実況対戦ポピュラス、祝一平VS西川善司の対戦予想コーナー。からくも西川氏の勝利に終わり，祝氏の**言いわけにも似た捨てゼリフ**で幕を閉じるでしょう（当たった？）。
千葉　仲(20)　宮城県

高いところで展開している。洪水対策か？
丹（また見に来ている）「いや，やりこんだ人なら洪水は使いません。火山を何発も起こしたほうが有効ですから」
　高い土地をいじるほうがマナがいるんですよね。マナの少ない序盤にこういうことをしていいのかなあ。
19:45　やはり人口比7:3ぐらいに差がついて，今度は祝氏が中央部を押さえた。苦しい善司くん手作りの山で反撃！　また泥沼の戦いが始まる。立てる崩す，立てる崩す，立てる崩す，沼にはまる。
善「やっぱり沼が奇跡のなかでは一番有効ですからね」
祝「えっ？　沼の弱点知らないの？」
善「……そんなこと言って動揺を誘おうとしてるんでしょ」
祝「そう思う？」
　直接対戦ならではの口頭の戦い。
20:17　祝氏がメガネをはずした。気合いの入れ直しか（どうでもいいが，氏はサングラスがとても似合うお方である）？
　お互いの境界にまんべんなく山が立っている。やはり山の被害のせいか，祝氏のリード幅が縮んだような。
「シュイィィン」あ，火山だ，祝氏が火山をおみまい！　さらにシンボルを移動にかかる。ここで一気に攻勢に出るのか。
祝「あれ，できない」
　リーダーは敵陣との境で死んでいた(笑)。
　善司くんは山を作って，相手の復旧の間に領地を広げる作戦に，祝氏はリーダーを誘導して個別撃破の作戦に出ています。
20:33　祝氏のリーダーは合体を繰り返し，パワーのある奴になりました。楽しげに誘導先を選ぶ祝氏ですがその途端……。

ゴボシ
祝「……！」
　リーダーのいたところには沼が広がっていた。ギャラリーが無責任に笑う。
祝「……（**シュイィィン**）」
　善司くんの領土に怒りの火山が炸裂！
21:10　そろそろ勢力が五分五分というところ。やはり善司くんは攻勢にたけています。おっと，何を考えたか善司くんが自分の領土に地震をしかけました。
善「こうやってシンボルに人を集めるんですよ」
　恐るべき早さで最強の騎士が誕生。さらに騎士が敵地に向かっている間にも手加減しない善司くん。
善「ああ，祝さんたら僕に無断でこんなところに城を（**カチカチカチカチ**）」
　山を立てている間に騎士が祝氏の領土に到着！　が，祝氏は慌てずに騎士の周りに穴を掘り，騎士を水の中に沈めてしまったぁ。もがく騎士。体力が少しずつ落ち始める。善司くんはぜんぜん気がついていない。ギャラリーは笑いたいのを必死にこらえています。そのまま何事もなかったかのように自分の領土を整備している祝氏。数分してふと善司くんが右上のウィンドウを見ると……。
善「ああっ，なんかもがいてるぅ」
　だはははははと爆笑するギャラリー。たちまち敵住民を池に落とすという「水攻め攻撃」が乱れ飛びました。
22:27　山を残しながら，自分の領地はしっかりキープしている２人。しかしやはり中心部は善司くんが取り，祝氏は周辺部に追われています。自分の領地に地震をしかけている善司くん。出てきた人間を，シンボルのある敵陣まっただなかに集合させる「一方的ハルマゲドン」攻撃です。騎士同様の追い込み技ですね。
　しかしそれでも事態は終結しない。千日戦争状態にあると判断した中野氏が，善司くんにハルマゲドンを起こすよう指導勧告。以後善司くんは奇跡を起こすのを控え目にして，マナの集積をはかる。一方祝氏は，再びマウスを汗だくでクリック。
祝「もーいや，こんな生活」
　挽回はできなかったが，この抵抗が効いて善司くんがハルマゲドンを起こすまでにはさらに１時間を要したのだった。
23:45　**「ウホウホウホ」**ハルマゲドンスタート。人口ゲージは祝氏の圧倒的不利を伝えている。ああ，やはり祝氏も善司くんの独走を止められなかったか。画面の中で２人のリーダーが向き合った瞬間！
ピタ！
　うぉぉ，ハングだあ！　天は祝氏に武士の情けをかけようというのかー！
　結局波乱のラストを乗り越えて，西川善司のハルマゲドン勝ちが決定しました。
　結局善司くんの２勝という形になりましたが，聞いたところでは祝氏は対戦がこれで３回目ということですから，いかに420面まで進んでいても，対戦ポピュラスのノウハウのある西川善司くんに一日の長があったといえるでしょう。
　しかし，この対戦もさらなる戦慄の歴史の序章に過ぎないのです。このあともさらに西川善司対中野修一などの数々の恐ろしい戦いが，編集室では繰り広げられています。対戦ポピュラスは確かに面白い。時間は使うし電話代もかかるし友人関係も下手するとこわれる。それでも対戦ポピュラスは面白い。あなたはこの面白さにつかってみる勇気がありますか？





X68000 10万台突破記念

愛読者特大モニタプレゼント

Oh!Xは通巻100号なんだよ～，とはしゃいでいたら，ほとんど時期を同じくしてわれらがX68000が10万台出荷を達成した。これぞ歓喜の2段重ね！　ここはひとつシャープさんにお願い！　というわけで豪華プレゼントを提供していただきました。どうです，スゴイでしょ。特に大型ディスプレイやカラーイメージスキャナなんて持っている人，少ないんじゃないかな。えっ，本体はないのかって？　だって大部分の皆さんはすでにX68000ユーザーじゃないですか。それに周辺機器ならX1/turboユーザーでも使えるでしょ。なに，X68000に乗り換えたい？　だったら本体ぐらい自分で買わなきゃね（というのがOh!Xの本音なのだ）。なお，9番以外はモニタプレゼントだから，当たった人には感想文をお願いしま～す。

1



21型カラーディスプレイ

CU-21HD　148,000円　1名

着脱可能なスピーカーを搭載した大きなカラーディスプレイ。

2 熱転写カラー漢字プリンタ

1名



3



CZ-8NS1　188,000円　1名

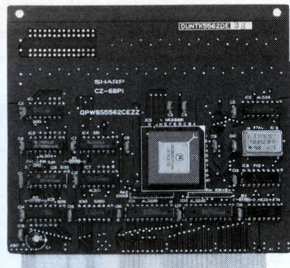
4 サイバースティック

CZ-8NJ2

1名



5 数値演算プロセッサボード



CZ-6BP1　79,800円　1名

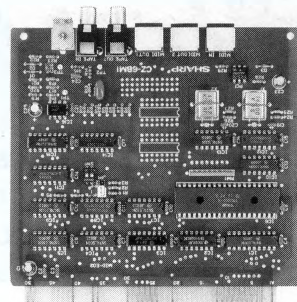
面倒な計算やレイトレーシング，シェーディングなどの処理速度を一気に高めることができるこのボード，CGには最適。

6 MIDIボード

CZ-6BM1

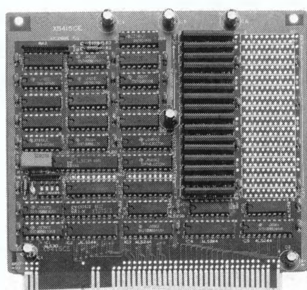
26,800円

1名



最近はいろいろなゲームもMIDI対応になっている。このボードがあればMIDI楽器が接続でき，鮮やかなサウンドが楽しめる。

7 2MB増設RAMボード



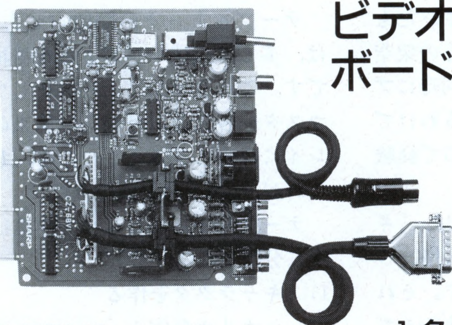
CZ-6BE2　79,800円　1名

あれもこれもパソコンでやりたい，という人はRAMボードの増設は必至。そんなあなたにこのボードをプレゼント。
（1M増設済のこと）

プレゼントの応募方法

とじ込みのアンケートはがき（ただし，今月のもの）の該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1990年8月18日の到着分までとします。当選者の発表は1990年10月号で行います。

8 ビデオボード



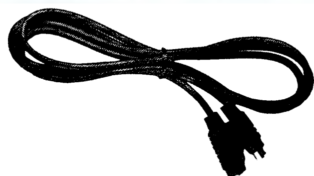
CZ-6BV1　21,000円　1名

このボードを使えば，X68000で作ったグラフィックや，プレイしているゲームなどが，簡単にビデオに録画できるようになるぞ。

（以上，シャープ提供）

9 キーボード延長ケーブル

1,980円



黒/グレイ
各5名

九十九電機より創刊100号を記念して，オリジナルのキーボード延長コードをプレゼント。寝っ転がってキーボードも打てるかな。

6月号プレゼント当選者

1 **A)ジェミニウイング**（沖縄県）宇良秀樹　**B)闇の血族**（静岡県）野村一洋　**2** **ねじ式**（千葉県）田浦達也（山口県）大隅研治　**3** **The File Professor**（東京都）高橋信博（静岡県）戸塚昭信　**4** **サイクロンEXPRESSα**（秋田県）佐々木仁志（神奈川県）鈴木利明（東京都）三田恭一郎（静岡県）三橋和美（大阪府）藤沢直樹　**5** **A)FAR SIDE MOON**（広島県）本谷正樹（愛媛県）横山智生　**B)A列車で行こうII**（富山県）加賀見政和（長崎県）佐藤充浩　**C)大海令**（埼玉県）桑原智志（岡山県）梅田敬　**D)南海の死闘**（東京都）小山薫（広島県）岸本秀生　**6** **クォース**（北海道）加納一郎（福島県）村上健（京都府）村久木康夫　**7** **ジャック・ニクラウス・テレフォンカード**（宮城県）伊藤洋美（東京都）平尾雄一（神奈川県）長嶺隆（奈良県）野瀬正博　林衛　**8** **スタークルーザー X68000用**（福島県）岩渕正樹（東京都）大橋飛雄吾　**XIturbo用**（神奈川県）田口聡（岡山県）小谷恒　**9** **キューブランナー**（東京都）角野俊人（神奈川県）武藤俊哉（京都府）田中啓　**10** **レナム**（岩手県）片岸健一（群馬県）石山篤志（兵庫県）郡茂樹（新潟県）霜鳥博史（香川県）佐竹勝博　**11** **A)ガンマ・プラネット**（東京都）高橋明（群馬県）藤田明（愛知県）永井周作　**B)グランディフロラム**（千葉県）久原義弘（栃木県）佐藤崇（三重県）大橋隆太郎　**C)Simple-CAD X68K**（福島県）仲山秀樹（和歌山県）辻本浩一　**12** **上海II**（長野県）吉沢克明（兵庫県）堀江良孝　**13** **ポピュラス**（千葉県）佐藤一成（島根県）原誠（鹿児島県）園田光太郎　**14** **プログラムオペレーティングシステム**（東京都）木部幸雄（石川県）川口聡　**15** **PIO-6BE1-A**（東京都）飯塚晃太郎　**16** **銀河英雄伝説＋set**（埼玉県）武藤一文　加藤勲（京都府）牧本隆　**17** **G68KII**（東京都）信川洋（福岡県）平山謙司（宮崎県）土井順之　**18** **A)D-RETURN**（神奈川県）細井実人（茨城県）伊東臣明　**B)ずるかまし**（宮城県）坂井一弘（東京都）千葉広道　**19** **A)オリジナルコーヒーカップ**（北海道）飯田伸一（愛知県）五月女優（広島県）田村和廣　**B)ツインビー**（茨城県）内田好則（京都府）上野政幸　**20** **バトルチェス**（三重県）水谷泰三　**21** **A)Zerø**（愛媛県）武智和彦（鹿児島県）本真光　**B)Misty3**（茨城県）地引秀和　原田大輔　**22** **セレクテッドソーサリアン　1**（長野県）塚本隆司（岡山県）横山博道（福岡県）浜地啓　**2**（東京都）松村一朗（神奈川県）三沢弘之（山梨県）深沢享広　**3**（茨城県）程田勝也（兵庫県）村上貴之（大阪府）中山良樹　**23** **ウインドブレーカー**（北海道）渋谷康則（東京都）八木貴弘（神奈川県）久崎圭（岐阜県）山口忠（大阪府）鈴木哲也　**24** **「この木なんの木」のCD**（茨城県）染谷祐一（福岡県）徳久雅人（大分県）山田博

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

（価格はすべて消費税別です）

ポケコンでCARPGを



Matsui Shin
松井　信

おっと，100号記念にちなんでポケコンの記事も復活かな？でも何をやるかというと，実はテーブルトークのRPGを楽しむのに利用しちゃおうというお話なんですね。使用するのは圧倒的シェアを誇るPC-E500シリーズです。お楽しみに。

CARPGとは，Computer Aided Role-playing-game，つまり，コンピュータを利用したRPGのことです。私がいま名づけました。コンピュータRPG（以下CRPG）ではありません。あくまでもテーブルトークRPG（以下テーブルトーク）のサポートを目的としています。

テーブルトークRPGとはなにか？

テーブルトークとは，机の上で多人数でやるコンピュータを使わないRPGです。というよりは，CPRGのほうをコンピュータ上でやるテーブルトークの真似ごとといったほうが正確です。

D&D(Dungeons&Dragons)などのテーブルトークは，最近になってようやくやっている人も増えてきたようですが，それでも実際にやったことのある人はまだ少ないようで，RPGといえばCRPGのようなゲームと思っている人も多いようです。しかし,CPRGはテーブルトークから戦闘システム部分とストーリー進行を抜きだしたもので，それはテーブルトークの楽しみのごく一部に過ぎません。

テーブルトークの楽しさとは基本的にロールプレイ，すなわち「ゴッコ遊び」の楽しさです。つまりRPGというからには,キャラクタを演じられることが必要です。

CRPGでは,キャラクタを動かしていこそすれ，演じているとはとうていいえません。ドラクエをしていて自分が(本当に)勇者だと思いながらやっている人はたぶんいないでしょう。

しかし，テーブルトークでは，あなたはガラスの仮面のごとく，完全にキャラクタになりきって，現実世界のようにファンタジーワールドの中を冒険することができるようになるのです。いくつかの作業と若干の想像力を必要としますが，こういったリアリティと面白さはCRPGの比ではありません。

テーブルトークの実際

とはいえ，テーブルトークにも問題点があります。ひとつは，1人ではできないという点，しかも，そのうちの1人は「マスター」と呼ばれる進行役にならなければいけません。そして，ある程度の時間（数時間以上）と，場所（人数＋机のスペース）が必要です。そういえば，マニュアルとそのほか道具も必要です。

テーブルトークはCRPGのように買ってきてすぐにできるものではありません。

とにかく，マスターになる人が，シナリオと呼ばれる台本（のようなもの。ゲームの設定およびストーリーなどを書いたもの）によって，ゲームを進行し，その架空世界のすべての出来事を管理し，同時にプレイヤーの不条理な要求に対処するわけです。当然，かなりな負担がかかるので経験者が望ましいわけです。

一方，1人ひとりのプレイヤーは，「キャラクタ」というゲーム上での仮人格，つまり，その世界での自分を持ちます。それには，強さ，魔法，持ち物，その他さまざまな属性が決められていて，その世界におけるキャラクタの個性を表し，その行動に一定の制限を与えます。この辺はCRPGと一緒ですが，CRPGでは戦闘に関係ない属性はほとんどないのに対して，テーブルトークには戦闘以外にもさまざまな属性が存在します。キャラクタというのはひとつの人格なのだから，これは当然でしょう。

以上，テーブルトークのいい点として，

1)　別人格を演じることができる
2)　実際にはない世界で遊ぶことができる
3)　破壊衝動(?)を満足でき，ミッションに成功したときはカタルシスが得られる

ということがあげられます。また，

4)　議論や会話の訓練になる
5)　多人数でわいわい遊べる
6)　マスターになって，いいシナリオができたときは自己顕示欲(?)を満足できる

などのメリットも忘れることができません。

これだけの利点を持つテーブルトークが，ボードゲーム界に与えた影響は大きく，SLGなどは駆逐されかかって，SLGの雑誌であったタクテクスなどは，本家が季刊になって，月刊のRPG雑誌を出しているほどです。

CARPGとは

前に述べたように，やはりマスターは大変です（同時にやりがいもあるが)。そうしたある日，疲れたマスターである私は，ひたすら作業をしていて思いました。

テーブルトークの問題点である「作業」は，多くは数値の処理という機械的な作業です。これをコンピュータ化してしまえば，マスターの負担は軽減し，本来のロールプレイに専念できるようになるんじゃないか。これが，CARPGなのです。

テーブルトークにおける作業は，次のように分類されます。

1)　キャラクタを作る
2)　シナリオを作る
3)　ゲームをする

まず，1)ですが，この辺は作業というよりは楽しみに属するものなので，ワープロの利用ぐらいにとどめておきます。

次に2)ですが，シナリオを作るというのは，小説のあらすじを作るようなものです。

まあ，仲間内でやるんだったらストーリーはどこかからパクってくればいいのですが，敵の設定，地図作成，ストーリーの記述といったところだけでもかなりの作業となります。

これは，市販のシナリオを買ってくればすむ問題ですが，何千円もする高いものだし，そんなにたくさん出ていません。それに，自作シナリオを成功させることこそがマスターの醍醐味だし。というわけで，この辺のCARPG化はそのうち取り上げた

いと思います。

そして，なんといってもマスターがいちばん大変なのは，3)の実際のゲーム中でしょう（と私は思う）。

なにしろプレイヤーは何人もいるのにマスターは１人なのだから。戦闘場面でたくさんの敵キャラクタを操りながら，プレイヤーの受け答えをするのは，やっぱり大変なことです。たとえば，

プレイヤーＡ：ゴブリン６に３ダメージ！
マスター：はぁい。
プレイヤーＢ：オーガ３に12ダメージ！
マスター：はいよ。
プレイヤーＣ：魔法かけるよぉ。ホールドパーソン！　ゴブリン４と５！
マスター：はーい（コロ，サイコロを振る）。５は止まった。それから？
プレイヤーＡ：そっちの番だよ。
マスター：そうか。じゃいくよ。ゴブリン１が，えーと誰の前？　あ，そう。アーマークラスいくつ？（コロ）当たった。えっとダメージは（コロ）２ね。じゃ，ゴブリン２は……。

これをえんえんと繰り返すのだから，慣れれば機械的にできるとはいえやっぱり面倒くさい。ましてや徹夜でやっていたりすると，うっかりするとパニックになりかねません。

そこで，戦闘中の敵モンスターのヒットポイントや攻撃を，コンピュータに管理させようというわけです。このプログラムを次回掲載する予定です。

コンピュータはなにを使う？

ところで，CARPGに使うコンピュータはなにがいいか。それは実はポケットコンピュータなのです。

まず，学校なんかでやるときは持ち運びができなくてはいけません。その点，ポケコンなら持ち運びもできるし，値段も安く，また，高級電卓として使えるので無駄な投資にはなりません。それに，工学系の大学生のほとんどはポケコンを持っているでしょう。

こういうと，ポケコンなんて，という人もいるかもしれませんが，今のポケコンをなめてはいけません。シャープのPC-E500（または，PC-1480U，PC-1490U）は，X1のBASICのような（というよりもN88-BASICのような）強力なBASIC，パソコンにも引けをとらない高速性，40×４行の広い画面，32KバイトのRAMは一部をRAMディスクとして使用でき，RS-232Cケーブルでパソコンにつなげる，などとてつもなく強力なマシンなのです。

では，自宅でやるならパソコンでいいやという意見もあるでしょうが，テーブルトークではマスターの情報はプレイヤーに見せてはいけないことになっています。したがって，机の上にマスターに向けてディスプレイが載ることになり，普通の家ではちょっと苦しいでしょう。そのため，ポケコンのほうが都合いいのです。

テーブルトークを始めるには

現在，たくさんの種類のテーブルトークが市販されていますが，やはりおすすめはD&Dおよび，AD&D(Advanced D&D)です。したがってこの連載も，対象は基本的にD&D，AD&Dとします。

D&Dはやはり日本ではもっともメジャーで，サプリメント（追加シナリオ，その他ゲーム補助用のツール）が多く，またルールがシンプルなため初心者でもやりやすいという特徴があります。

しかし，実は米英ではAD&Dのほうが遙かにメジャーで，そのサプリメントの量はD&Dの比ではありません。ルールもD&Dより体系化され，より面白くなっています。なにぶん英語というハンデがありますが，高校生でも読める程度のものですからそれほど心配することはありません。日本語版も７月から出版されるはずですが，最初は誤植が多いと予想されるので，いっそのこと英語版を買っても無駄にはならないでしょう。

というわけで，ようやくテーブルトークを始めるわけですが，なにもしたことがない人がいきなりマスターを始めるのは大変です。しかし，誰かがマスターをやらなければいけません。しかし，なにかとんでもない間違いをする可能性もあります。

そのため，まず最初は（少なくともマスターをやる人は）どこかでテーブルトークを体験してくることをおすすめします。たとえば，どこでも高校，大学なら誰かしらはテーブルトークをしているものですから，友達のつてから仲間に入ってみるというのがひとつの手です。

ほかに，テーブルトーク関係の雑誌には地域的なテーブルトークサークルのメンバー募集が出てますからそこに連絡を取ってみるという手もあります。

マスターへの道

とにかく，マスターになる人はロールプレイとはなにかということを理解しなくてはいけません。ルールを読むことも忘れずに。

それから，特にファンタジー系テーブルトークの場合，たくさんのファンタジー小説と，ヨーロッパの歴史書，そしてそれ以外にもたくさん小説も読んで素養をつけておきましょう。マスターというのは，作家にして脚本家，監督にして俳優というとってもやりがいのある総合プロデューサーなのです。

それでは来月はプログラムに入ります。

〈参考文献〉
Ｄ＆Ｄがよくわかる本　富士見文庫　490円
D&Dが具体的にどのような手順で進められるのかわかる。D&D初心者にはおすすめ。
眠れる龍　現代教養文庫　720円
アメリカのゲーマーの生活がわかってなんとなくほのぼのする。ファンタジー小説としてもいい出来。
ドリームパーク　創元推理文庫(SF)　580円
マスターの内輪うけと評されるだけあり，マスターをやっている人には面白い。

ゲーム紹介(1)

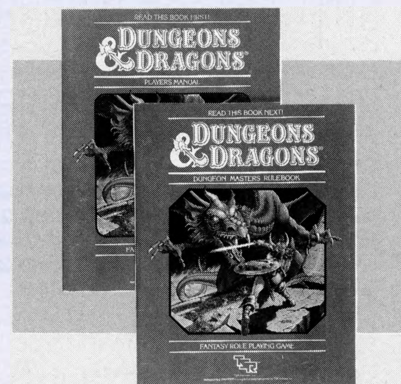
Dungeons & Dragons

RTS.inc.（日本語版：新和）

テーブルトークといえばD&Dというぐらいメジャーなゲームで，特に日本ではほぼ主流となっている。とにかくルールが簡単で覚えやすく，初心者でもとりあえず20面サイコロを振って殴っているだけで十分楽しい。

しかしながら，古いゲームであるということは否定できず，攻撃は最大の防御でありキャラクタのレベルが２桁になるころからなにか間違ったゲームへと発散していく傾向が多々ある。最高レベルである36のあとには，みんなで神様をやろうというルールまであるが，きっとただの冗談だろう。

通称，赤Ｄといわれるベーシックと，青Ｄといわれるエキスパートの２つの箱が最低限必要。



上級セットとしてコンパニオン，マスター，インモータルの拡張ルールセットがある。

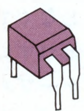
ハードウェア工作入門《2》

基本インタフェイス回路 その2

Misawa Kazuhiko
三沢 和彦

今回は製作実習編です。とても簡単な回路ですし，実体配線図も用意しました。注意事項も徹底的に詳しく解説してありますから皆さんも部品を揃えて実際に挑戦してください。うまくいったときの喜びは格別ですよ。

いよいよお待ちかねの製作実習編です。今月が待ち切れなくて，もう部品を揃えてしまった人もいるかもしれませんね。とにかく，まずは部品表のとおりに部品を揃えてください。



汎用ケーブルの製作

最初にジョイスティックポートと自作回路とをつなぐためのケーブルを作ります。このケーブルは1本作れば，連載で製作する回路すべてに使えるようにしてあります。圧着用の10ピンフラットケーブルとコネクタとは買ったお店で圧着してもらっておきます。部品を買うときに頼めば，その場で圧着してくれるはずです。

圧着されたコネクタを見ると，一番端に印がついているでしょう。これが1番ピンです。さて，このフラットケーブルを9ピンDサブコネクタにハンダ付けしていきます。9ピンDサブコネクタはメスコネクタでなければ，X68000につなげないので注意してください。

そして，Dサブコネクタの表に出る側をよく見ると，小さく1〜9の数字が記されているのがわかるでしょう。そこで，圧着コネクタの1番ピンにつながっている線から順番にDサブコネクタの各端子にハンダ付けしていくのです。Dサブコネクタの端子どうしの間隔が意外と狭いので，ハンダが隣とくっつかないように注意してください。

ところで，ケーブルは10ピンでDサブコネクタは9ピンですから1本余ることになりますが，10番の線は9番ピンのGNDにいっしょにつないでおきます。ハンダ付けが無事終わったら10本のケーブルを束ねてDサブコネクタケースについている金具で止め，コネクタ全体をケースに納めます。これで出来上がり。



基本I/O基板の製作

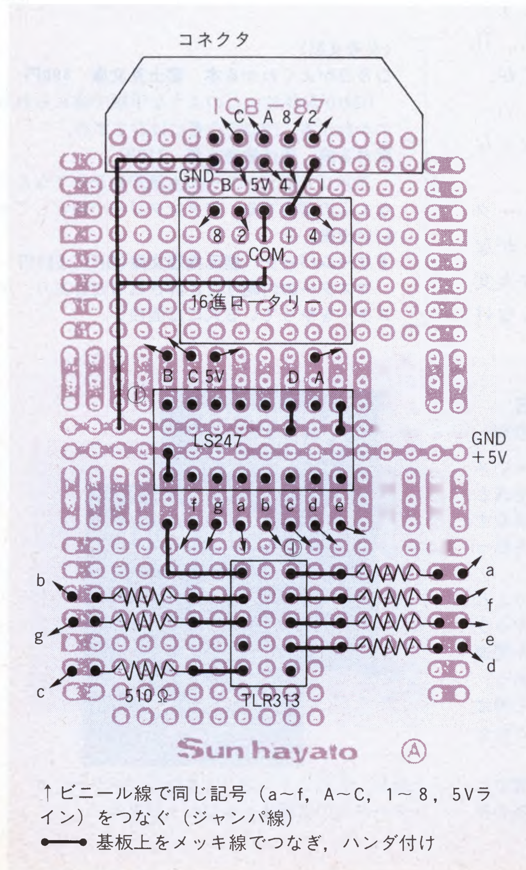
基板上に回路を組むときにもっとも頭を悩ませるのが，部品の配置です。部品の配置をうまく決めるかどうかで配線の手間がまったく違います。皆さんは図1の実体配線図を参考にしながら，以下の説明を読んでください。

サンハヤトのICB-87という基板はIC1個用の汎用基板で，ICの足まわりの配線がしやすいように工夫されているものです。次回に製作するA/Dコンバータもこの基板上に作るので，何枚かまとめて買っておくのもよいでしょう。

●主な部品の取り付け

まず最初に，先ほど作った汎用ケーブルをつなぐ基板用コネクタを取り付けます。まずは10ピン全部をハンダ付けしてしまいます。このとき，ハンダ付け面から見て，ジョイスティックコネクタのピン番号は図2のように対応しています。ハンダ付けしたピンから各場所への配線はまだ行いま

図1 実体配線図



↑ビニール線で同じ記号（a〜f，A〜C，1〜8，5Vライン）をつなぐ（ジャンパ線）
━━ 基板上をメッキ線でつなぎ，ハンダ付け

部品表

9ピンDサブメスコネクタ	1個	200円
Dサブコネクタケース（DE-C1-J6）	1個	360円
10ピンフラットケーブル	1m	100円
10ピンコネクタ（PS-SRN10）	1個	200円
IC用基板（サンハヤトICB-87）	1枚	90円
10ピン基板用コネクタ（HIF3BA10P-DS）	1個	100円
16進ロータリースイッチ（アルプスSRRQ）	1個	250円
ICソケット16ピン	1個	35円
74LS247	1個	80円
TLR313	1個	210円
抵抗510Ω	6本	10円
ビニール配線材	少々	

図2 基板用コネクタ(ハンダ付け面から見た図)

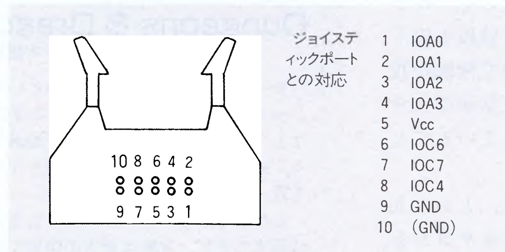
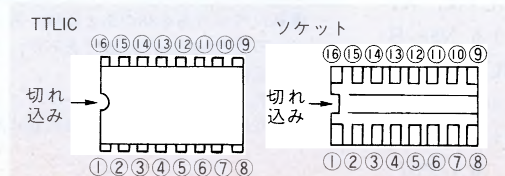


図3 IC，ソケットを上から見た図



せん。

次に，ICソケットを取り付けますが，ソケットを上からよく見ると図3のように片方に切れ込みがあり，これを目印にピン番号が決まっています。

規格表などに載っているICのピン番号はICの上から見たときのものなので注意してください。当然，配線している側から見ると逆回りになっています。この点は熟練者でも意外と勘違いすることがありますので油断しないように。もちろんこの連載では，実体配線図に従えばOKです。

次に，ICソケットを基板に差し込んだら，すぐに8番ピンと16番ピンを内側に折り込んでハンダ付けしてしまいます。というのも16ピンICの場合は8番がGND，16番が5Vに接続するのが一般的だからです。そして，基板がICB-87の場合は実体配線図を見てもわかるとおりICの2列の足の間にGNDラインと5Vラインの2本の配線ラインが通っているので，折り込んだ8番ピンと16番ピンとをそれぞれそのラインにもハンダ付けします。

このようにIC工作では，GNDラインと5Vラインを先に通してしまうのが基本なのです。これができればあとはICの足1本1本をすべてハンダ付けしていきます。今回は6番ピンもGNDに落とすので，内側に折り込んでGNDラインにハンダ付けします。

次に7セグメントLED（TLR313）をハンダ付けします。これはソケットがないので直接基板にハンダ付けしてしまうしかありません。TLR313のピン番号は先月号にも載せてありますが，やはりハンダ付けする側から見ると逆回りになっていることに注意しましょう。

TLR313の10番ピンは5Vラインに直結ですが，1〜4，6，8，9番ピンは510Ωの抵抗を介してLS247につなぐので，次に抵抗の配線を行うのが効率的です。配線の都合上，抵抗は実体配線図のように寝かして差し込み，TLR313側の足は折り曲げて，図4のようにTLR313の各端子まで伸ばしてハンダ付けしてやります。反対側は基板の端に並んでいる端子にハンダ付けしてやり，余った長さはすっぱり切り落としてしまいましょう。

こうして7本の抵抗を付け終えたら，16進ロータリースイッチを取り付けます。私の手に入れたアルプス製のものは取り付け用の足も端子も位置としてはIC用基板に適したものですが，ただひとつ取り付け足が大きすぎて，基板の穴にはそのままでは入りません。そこで，錐（きり）を使って取り付け位置の穴を少し大きくしてからはめ込みます。はめ込んだら端子をハンダ付けしてしまいましょう。

ここまでくると，部品はすべて取り付けられたことになります。ここでセンスの鋭い人はお気づきでしょうが，工作では，配線の前にすべての部品を取り付けてしまうのが鉄則です。それは何度もいうように，部品の配置とバランスが工作の手間を決めているといえるからです。

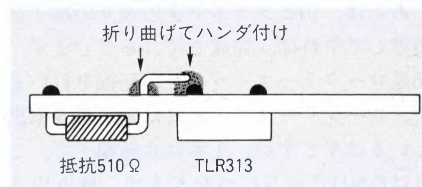
●部品間の配線作業

部品がすべて配置されたら，次は地道に配線作業です。まずは抵抗7本とICとの接続をしましょう。TLR313のa〜gから伸びている各抵抗の端をLS247のa〜gに対応させて，被覆されたビニール線でつないでいきます。このようにつなぎたいところどうしをジャンプしてつないでいる線のことをジャンパ線といいます。

このとき，TLR313のa〜gの並び方も間違えやすいですし，そのうえ，LS247のa〜gが9〜15番ピンに割り当てられていますが，これも順番に並んでいないので注意が必要です。実は，私も最初は間違えてつないでしまいました。間違えてつなぐとLEDの表示がおかしくなりますが，壊れることはありません。それから，10番ピンを5Vラインにつなぐのも忘れないように。以上で，LEDまわりの配線は完了です。

次にLS247の入力1，2，7番ピンの配線です。ここは，10ピンコネクタに直結しますが，10ピンコネクタのピン番号も間違いやすいので，再度図2を確認してください。このピンは位置も連載のすべての回路に共通です。

図4　TLR313と抵抗のハンダ付け



ところで，今回の製作でいちばん難しいのがこの10ピンコネクタまわりの配線でしょう。隣と近いうえ，ビニール線がかさばるので，次に述べるように手際よく行います。まず，コネクタ側の端子はあらかじめハンダ付けしておくこと。そして，ビニール線の被覆を必要な分（1mmほどで十分）だけワイヤストリッパでむいておき，そこにもハンダを付けておきます。

このように，ハンダ付けする両側にあらかじめハンダを付けておくのがコツです。あとは，ハンダゴテを基板側に当て，ビニール線の先をハンダ付けしたい箇所につけるだけで意外とうまくできます。万一隣にもくっついてしまった場合には，ハンダ吸い取り器で完全にハンダを取り除き，最初からやり直します。一度失敗したハンダは，二度とくっつかないことを肝に命じておく必要があります。

最後に3番ピンを5Vラインにつないで ICまわりの配線も終わりです。4，5番ピンはなにもしないでおきます。参考までに3〜5番ピンの機能を囲み記事の中に記しておきますので，なにか自分で設計工作す

抵抗のカラーコード

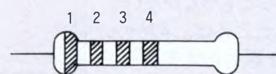
抵抗1本1本をよく見ると色のついた4本の帯が見えますが，これが抵抗値を示すカラーコードです。精度の高い特別な抵抗には5本ついているものもありますが，ここでは一般によくみかける4本組の読み方を説明します。

最初の3本が抵抗値そのものを示し，最後の1本は抵抗の精度を示しています。精度というのは，表示されている値を基準にして実際の抵抗値にどれだけ誤差があるかということです。たとえばそれが金色のカラーコードであれば，実際の抵抗値は表示値の±5％という意味ですから，100Ωの抵抗の場合なら，実際は95〜105Ωになっています。

最初の3本の見方をマスターしましょう。そこは0〜9の9種類の色で表されていて，1本目と2本目とで2桁の値を示し，3本目でさらに10の何乗倍かを示します。図中の例題で確認してください。皆さんは，0〜9が何色に対応するか覚えましょう。それには，0から9までならべて「くちあだき，みあむはし」と語呂で覚えます。それぞれの色名の頭文字を並べただけですが，なかなか覚えやすいと思います。

	1本目	2本目	3本目	4本目
黒(く)	0	0	10の0乗＝1	
茶(ち)	1	1	1 ＝10	±1％
赤(あ)	2	2	2 ＝100	±2％
橙(だ)	3	3	3 ＝1,000	
黄(き)	4	4	4 ＝10,000	
緑(み)	5	5	5 ＝100,000	
青(あ)	6	6	6 ＝1,000,000	
紫(む)	7	7	7 ＝10,000,000	
灰(は)	8	8	8 ＝100,000,000	
白(し)	9	9	9 ＝1,000,000,000	
金			10の−1乗＝0.1	±5％
銀			−2 ＝0.01	±10％

カラーコードの位置



1　2　3　4
例1　茶　黒　赤　金
1　0　$\times10^2$　＝1000Ω（1kΩ）
例2　緑　茶　茶　金
5　1　$\times10^1$　＝510Ω（今回使っているもの）

るときの参考にしてください。

あとは，10ピンコネクタの残りの端子を処理してやれば，完成です。そこでまず，16進ロータリースイッチとの配線を行います。ロータリースイッチには端子が5本出ているはずですが，1本は共通端子で，これはGNDラインにつなぎます。残りの4ビット端子はどの順に最下位ビットから並んでいるかあらかじめチェックしておいてから，10ピンコネクタの1～4番端子につなぎます。

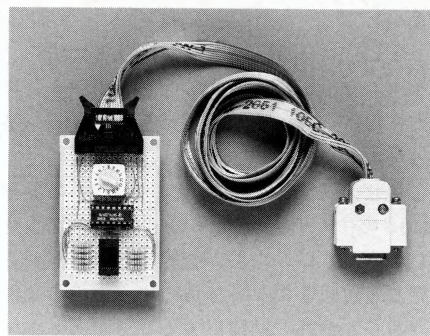
店で品物を買うときに各端子の機能を尋ねておくのが得策です。自分で調べることになってしまったら，まずロータリースイッチを1に合わせておいて5本のうちどの2本が導通しているかをテスターで計り，次に2，4，8と順次合わせて，やはりどの2本が導通しているか調べます。1，2，4，8すべての場合に共通な端子がGNDにつながり，あとはそれぞれどのビットに対応するかチェックします。

こうして，下位ビットから順に1～4ピンにつなげばOKです。実体配線図では，下位ビットから順に1，2，4，8，COMと記号を打ってありますが，品物によって位置が変わるかもしれません。最後に，10ピンコネクタの5番と5Vライン，9番とGNDラインとをつなぎます。この5VとGNDとを逆にするとICが死ぬこともあり得ますから，気をつけてください。

●完成後のチェック

以上ですべての配線が終了し，いよいよ完成です。配線が終わったら，実際にX68000につなぐ前にもう一度実体配線図と比べて，配線のチェックをしてください。ただし，一度配線が終わってからチェックするまでの間にはお茶を飲むなりゲームをするなり，なにか気分転換をすることが大切です。最初は必ずどこか配線ミスをしているものですが，これを発見するためには頭を冷やしたあとのほうがずっと効率がよいのです。

十分チェックしたら汎用ケーブルと基板のコネクタをつなぎます。コネクタには片側に出っ張りがあり，これで上下の向きが決まっていますので，向きに注意しながらしっかり差し込んで最後にフックで挟み込んで止めます。そしていよいよX68000のジョイスティックポート1に差し込んでみましょう。


これで完成だ！

どうですか，LEDに3が表示されましたか？　もし3が表示されなければ，まだどこかにミスがあります。ただし，このテストは必ずX68000の起動直後にX-BASICを立ち上げて行ってください。

とりあえず，あり得るミスについて考えてみましょう。

1)　なにも表示されない場合

5VラインとGNDラインの配線ミスです。単にどこかの配線し忘れか，もしくは5VとGNDとを逆につないでしまっているかもしれません。

2)　LEDは点灯するが，表示がおかしい

TLR313のａ～ｇとLS247のａ～ｇとの対応がきちんとなっていない。あるいは隣どうしのピンがショートしていることもあり得ます。

3)　表示はするが，3でない場合

Dサブコネクタか基板の10ピンコネクタまわりの配線ミス。LEDまわりの配線はOKです。

以上，どうしても配線ミスが見つからなければ，ICが死んでいることも考えられますが，実際のところICの不良は万に一つしかないと思って差し支えありません。根気よくミスを探してください。

＊

いかがでしたか？　まったく初めて工作する人でもこの程度の回路なら十分ついて行けるのではないでしょうか。完成したらさっそく X68000 からコントロールしてみたいところですが，はやる気持ちを抑えて次回までのお楽しみとしましょう。

来月はまず，ソフトウェアで最も基本となるI/OコントロールドライバをX-BASICの外部関数の形で提供します。といってもたいして難しくないプログラムです。最初は68000アセンブラ入門みたいな解説になるでしょう。そのあとにそのドライバを使った応用プログラムを作ってみます。同時に一般的なI/Oコントロールを行うための基本もきっちり押さえる予定ですのでお楽しみに。ではまた，来月。

LS274の機能

図は規格表からの抜粋です。この図を見ながら各ピンの機能を順番に説明しましょう。

●電源系統

まず＋5VとGNDは問題ないと思います。

●出力

出力（9～15番ピン）は7セグメントLEDのａ～ｇに対応して，抵抗を介して接続します。図中に小さくａ～ｇが書かれているのがわかるでしょうか。

●入力

入力は上位ビットからDCBAの順になっています。4ビット入力なので，0～15まで入力できますが，10以上になると意味のない表示になってしまいます。また，このICは表示の機能しかないので，たとえば桁上がりを自動的に足し込むようなことはできません。

●オプション

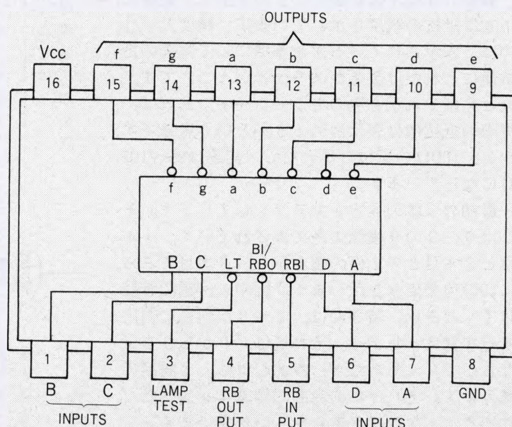
3番ピンはランプテストといって，ここをGNDに落とすと強制的にすべてのセグメントを点灯させます。通常は5Vラインにつないでおきます。

4番ピンのRBOと5番ピンのRBIはリップルブランキング機能に使うもので，通常はやはりなにもつながないでおきます。これはたとえば，4桁のLEDに3桁の数字を表示させるとき，上の桁の0を表示させないようにするために使います。

それには，最上位桁のRBIをGNDにおとし，そこから順に上の桁のRBOを次の桁のRBIにつないでいきます。RBIがLのときは，もし入力が0ならばなにも表示せず，しかもRBOからLを出します。

上の桁から順に0のときだけ数珠つなぎでなにも表示させないようにRBOをLにして伝達していきますが，ある桁で0でないときはそこから先はRBOがHのまま伝わっていくので途中の桁が0であっても0を表示します。

言葉で書くとちょっとわかりづらいかもしれませんが，もし電卓や時計などで数桁にわたって表示させたいときにはそれぞれの用途で専用の表示用ICが簡単に手に入りますので，LS247のこの機能について理解しなくてもかまいません。それでも興味ある人は各自規格表を見て自由研究としてください。



超入門・ファイル処理

Izumi Daisuke 泉　大介

このところ難易度の高くなってきた調理実習ですが，今回は基本にかえって簡単なファイル処理の方法を解説しましょう。また，応用としてYETのスコアファイルを複数のプレイヤーで使用するためのアレンジも行っています。挑戦してください。

ゲーム作りもひと段落ついたところで，今月はちょっと実務っぽくファイル処理に取り組んでみたいと思います。一般にファイルというと，書類を綴じ込んだものを指しますが，コンピュータの世界ではディスクに保存されているBASICのプログラムやワープロの文書，表集計ソフトやデータベースのデータなどのことをいいます。文字やデータを綴じ込んだものだと見れば，なるほどファイルと呼ばれるのもうなずけるような気がします。この対比でいくと，ディスクドライブはさしずめファイルキャビネットというところでしょうか。

これらのファイルの内容に対するさまざまな作業がファイル処理で，簡単なところではファイルの中から単語を検索し，その単語が含まれている行だけを抜き出す。ファイルに入っている文字数，単語数，行数を数える，といった作業があります。ファイル処理というと難しそうな印象を持たれるかもしれませんが，コツをつかんでしまえば実に簡単なものなのです。なんせあのC言語では入門編で取り上げられる程度の題材なのですから。

ファイルのオープン，クローズ

ファイルを扱うときの儀式として，ファイルのオープン，クローズという作業があります。紙綴りファイルから必要な情報を探し出すときにはファイルを開きますね，また，作業が終わればファイルを閉じてキャビネットに戻します。これに対応するのがファイルのオープン/クローズです。ディスク上のファイルに開くも閉じるもないような気がしますが，コンピュータにとっては別の意味を持っています。

ファイルをオープンするとは，このファイルを使っているよとコンピュータに宣言する作業です[1]。これによってコンピュータはそのファイルが使用中であると認識し，ほかの人が同じファイルを使おうとするとエラーを出すことができるようになります。ファイルキャビネットなら使用中のファイルはキャビネット内にはありませんが，ディスクでは使っていようがいまいが常にキャビネット内にファイルがあるようなものですからね。さらに進めて，見るだけなら何人の人が同時に見ようとディスク上のファイルが変更される心配はありませんから，複数の人がオープンできるようにすることもできます[2]。

逆に，ファイルをクローズするというのは自分が使い終わったことをコンピュータに知らせるための作業です。X-BASICを終了するとオープンされたままのファイルは自動的にクローズされますが，自分でオープンしたファイルは必ず自分でクローズするようにしたいものです。

●見るのか，更新するのか，作るのか

ファイルを使うといってもいろいろあります。ファイルをオープンするときには，そのファイルを見るだけなのか書き込みをするのか，すでに存在するファイルを扱うのか新たに作るのかを明確にしなければなりません。これが「アクセスモード」あるいは単に「モード」と呼ばれるものです。

X-BASICでは「読む」「書く」「読み書きする」「新たに作る」という4つのモードでファイルをオープンすることができます。ワープロならば最初に文書ファイルを読むだけ読んで，変更が終わったあとに今度はすべて書き出せばOKですが，随時データの読み書きが行われるデータベースではファイルは「読み書き」モードでオープンする必要があります。メモリに入りきらないほどの大きなデータベースもあります。こうなると「読み書き」以外には扱う方法がありません。

●ファイル番号でファイルを管理

X-BASICでファイルをオープンするときには，

　fopen（ファイル名, モード）

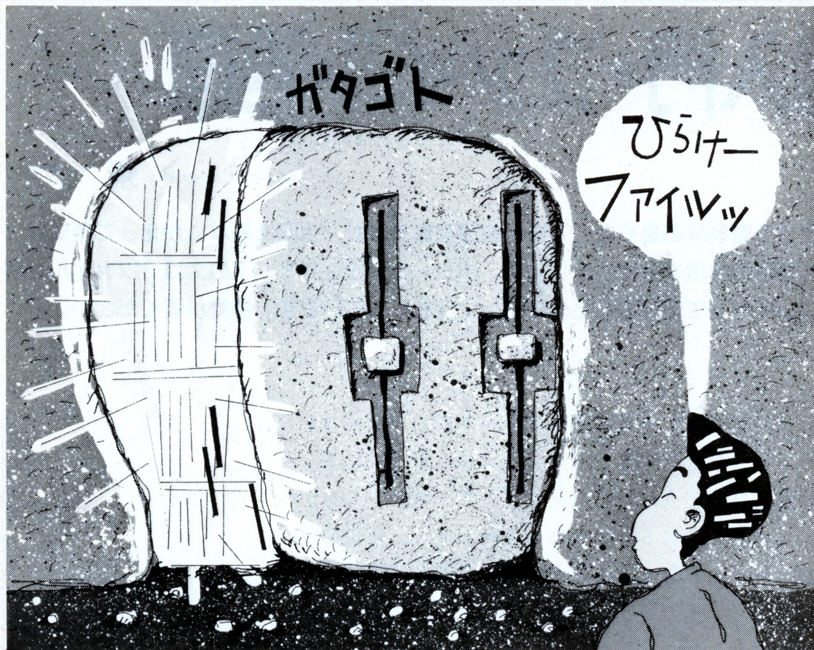
とします。モードはr（読む），w（書く），rw（読み書き），c（作る）と文字で指定するようになっています。たとえば，myfileというファイルを作りたいのなら，

　fopen(“myfile”,“c”)

となります。

ファイル処理を考えると，ファイルをひとつだけ

1)　ファイル管理を一手に引き受けているのはOS（X68000ではHuman68k）ですので，ここは正確にはHuman68kに宣言する，となります。
2)　マルチユーザーのOSではこの機能は必須といえるでしょう。X-BASICではファイルがすでにオープンされているかどうかのチェックすら行いませんが……。



しかオープンしないというのは稀です。あるファイルから特定の文字を探し出し，その文字列を含む行を別のファイルに書き出すというように，2つあるいは3つのファイルを同時にオープンして使うのが普通です。

ファイルがひとつだけならデータを読み込む，書き出す対象がどのファイルなのか迷うことはありません。しかし，オープンされているファイルが複数になると，対象がどのファイルなのか特定できなくなります。読み書きのたびにファイル名を指定するというのもひとつの解決法ですが，プログラムを書くのが面倒ですし，さらに1文字書くたびにファイル名の比較をやって対象のファイルを特定することになるので時間がかかってしかたありません。

そこでファイル番号[3]の登場です。オープンしたファイルに番号を付けておいて，あとはこの番号を利用して読み書きを行おうというものです。X-BASICではファイル番号はファイルをオープンしたときにfopen関数の戻り値として返される整数です。次の命令を試してみてください。

```
print fopen("test","c")
```

これでtestというファイルが新たに作成され，返されたファイル番号が画面に表示されるはずです。ファイル作成を指定すると，すでに存在するtestというファイルを消去して新たに作ってしまいますので注意してください。実際にはこのファイル番号を変数に入れておきます。

```
int file
```

と宣言し，表示された値を代入しておきましょう。

続けてもうひとつファイルを作ってみます。

```
int file2
file2=fopen("test2","c")
```

変数file2を表示して，返されたファイル番号を確かめてみましょう。

オープンしたあとのファイル操作はすべてファイル番号を使うと説明しました。ファイルのクローズも例外ではありません。クローズにはfclose関数を使い，引数にファイル番号を指定します。

```
fclose(file2)
```

なら，test2がクローズされます。もちろんtestはまだオープンされたままです。fclose関数はファイルを個々にクローズするのに便利な関数です。クローズ用の関数にはもうひとつfcloseallがあります。これはオープンされているファイルをすべてクローズするので楽ちんです。では，次に進む前にfcloseall関数でファイルを全部閉じておくことにします。

```
fcloseall( )
```

と入力すれば現在オープンされているファイルtestも（もしmyfileをオープンしているならそれも）クローズされます。

データを読み込んでみよう

さあ，いまや皆さんはファイルを開けたり閉じたりする方法を修得したわけです。fopen，fcloseという2つの関数はファイルの世界に入る最も基本的な呪文です。覚えた呪文はすぐに使って慣れるのがマジックポイント向上の秘訣とばかりに，さっそくfilesコマンドで表示されるファイルを片っ端から読み出しモードでオープンしている方もいらっしゃることでしょう。

そんな向上心旺盛なあなたに質問です。オープンできないファイルはありましたか？ X-BASICの世界，すなわちHuman68kの世界にはこの方法でオープンできないファイルは存在しません。どんなファイルでも（それがfilesコマンドで表示されるファイルなら）オープンすることができるのです。ワープロの文書ファイルやBASICで作ったプログラムのファイルはもとより，皆さんが使っているX-BASICもオープン可能です[4]。ワープロの文書やBASICのプログラムファイルは文字の集まりです。これに対しX-BASICはマシン語で書かれたBASIC本体です。X1のHuBASICなどではこういったマシン語プログラムファイルはオープンすることができませんでしたが，X-BASIC（すなわちHuman68k）ではなんの制限もありません。文字が収められたファイルも実際にディスク上ではASCIIコードの集まりです。つまり1文字単位で読み込めば，0～255の数値が返ってくるだけなのです。ダンプリストでお馴染みのマシン語は16進数2桁（これも0

3) Human68kではファイルハンドルと呼んでいます。またファイルを指し示すものという意味でファイルポインタと呼ぶ場合も多々あります。ファイル番号を保持する変数に fp という名前が多いのはこのファイルポインタを略したものです。
4) X-BASIC は BASIC.X というファイル名で BASIC2 ディレクトリ（あるいは BASIC ディレクトリ）に入っています。

～255の数値）の集まりですから，ファイル内では両者はまったく同じものだといえます。どんなファイルでもオープンできるというのはX-BASICのファイル処理の大きな特長です。

●１文字単位で読み込む

では実際にファイルからデータを読み込んでみることにしましょう。まずは適当なプログラムを作り，それをTEST.BASというファイル名でセーブしてください。以前作ったプログラムがある方はそれを使って結構です。

まずはファイルのオープンです。データを読み込むのですから“r”でオープンします。

```
int file
file=fopen(“TEST.BAS”,“r”)
```

ですね。

さて１文字単位の読み込みですが，これにはfgetcという関数を使います。ｃはcharacterを意味しています。先ほど触れたように，この関数は文字を返すのではなく，ASCIIコードを返してきます。

```
print fgetc(file)
```

を実行してみてください。先ほど適当に作ってセーブしたプログラムの最初の文字のASCIIコードが表示されます。行番号の前にはスペースが詰まっていますから，スペースのASCIIコード32（20H）が画面に表示されたはずです。このままではわかりづらいので，chr$関数でASCIIコードを文字に変換することにしましょう。これは，

```
print chr$(fgetc(file))
```

でOKですね。ファイルの最後まで続けて表示するのなら，

```
while 1:print chr$(fgetc(file));:
  endwhile
```

となります。セーブしたプログラムが表示され始めましたね。プログラムの最後まで表示すると……

「ピッ！（エラー音）」

ハイ，エラーです。

エラーが発生してしまいました（たぶん「バイトの範囲を越えました」と表示されているはず)。表示されたエラーメッセージを見てもなにが起こったのかわからないでしょうから解説しましょう。これはファイルの最後まで到達したにもかかわらず，さらにデータを読み込もうとしたのが原因です。ファイルの最後まで達すると，fgetc関数は－1を返します。

```
print fgetc(file)
```

として試してみましょう。ところがchr$関数はchar型の引数（0～255）しか受け付けません。つまりchr$(－1)を実行したのと同じことになりエラーが出たのです。ファイルを最後まで読み込んだら，それ以上読みに行かないようにプログラムする必要があります。

ファイルの最後に到達したかどうかを調べるにはfeofという関数を使います。この関数は，

```
feof（ファイル番号）
```

という書式で利用し，指定されたファイルが最後まで（end of fileまで）達していたら－1を，まだ達していなかったら０を返します。これを使って，

```
while feof(file) <> －1 : ～ : endwhile
```

と先のwhileループを書き直せば，ファイルの最後まで文字を表示し続けることができます。fcloseall関数でTEST.BASファイルをクローズし，もう一度ファイルのオープンからトライしてみましょう。今度はエラーも起こりませんね。最後に，

```
fclose(file)
```

でTEST.BASをクローズすれば，ファイル処理入門はめでたく終了です。

リスト１はマシン語ファイルを表示するためのプログラムです。マシン語ファイルはchr$で変換しても意味のある文字にはなりませんから，ダンプリストにならって２桁の16進数で表示することにしました。また数値がずらずらと並んでいるだけというのは見苦しいので，データ16個ごとに改行するようにしてあります。while～endwhileループでファイルエンドまで回しながら，for～nextを使って16個のデータを表示するという方法でプログラムしました。基本的には上の文字表示のプログラムと同じですからすぐにわかると思います。

このプログラムを使って，TEST.BASを表示してみましょう。２桁の16進数がずらずらと表示され，なにが入っているのかさっぱりわからないかもしれませんが，注意して見るところどころに「0D 0A」というデータが入っているのがわかると思います。この２つのデータは改行を意味し，プログラムをロードするときX-BASICはこのデータを手掛かりに行の終わりを判定しているのです。

リスト１　マシン語ファイルを見る

```
 10 str filename               /* ファイル名
 20 int file, data             /* ファイル番号、データ
 30 int readingFlag=1          /* 読み込み中フラグ
 40 int i
 50 /*
 60 input "ファイル名 : ",filename  /* ファイル名入力
 70 file=fopen( filename, "r" )   /* ファイルオープン
 80 while readingFlag         /* 読み込み中は以下を実行
 90   for i=1 to 16                 /* 16回繰り返す
100     if feof( file ) = -1 then {
110       readingFlag = 0 /* ファイルエンドならフラグ
120       break           /* をクリアしてループ中断
130     }
140     data=fgetc( file )     /* データを１つ読み込み
150     print hexStr( data );" ";       /* 16進で表示
160   next
165   print                           /* 次の行へ
170 endwhile
180 fclose( file )                 /* ファイルを閉じて
190 end                                     /* 終了
200 /*
210 func str hexStr( data )    /* 16進2桁の文字にする
220   return( right$( "0"+hex$(data), 2 ))
230 endfunc
```

システムディスクのBINディレクトリにはDUMP.Xというプログラムが入っています。これは

69 6E 70 75 74 20 22…… input ”……

というように，16進数とそれをASCIIコードと見なしたときの対応する文字を表示してくれます。リスト1はこの左半分だけを表示するようなものです。リスト1を改造し，DUMP.Xのような出力ができるように挑戦してみてください。

●データを読み込むそのほかの関数たち

X-BASICではfgetcのほか，freads，freadの2つの関数でデータをファイルから読み込むことができます。freadsは文字が入っているファイルを対象とし，改行コードまでの1行を一気に文字変数に読み込む関数です。1文字1文字読み込むより一気に読むほうが速いので，文字ファイル処理では多用される関数です。もう一方のfreadは1次元の数値型配列を一気にファイルから読み込む関数です。

●データを書き出す

ファイルにデータを書き出すときには，“w”モードか“c”モードでファイルをオープンし，データ書き出し用の関数を使うだけで基本的な作業はまったく同じです。データ書き出し用に用意されている関数はfputc，fwrites，fwriteの3つで，これまでに紹介してきた読み込み用関数と対になっています。

fwriteは実験データなどを1次元の配列に収めておき，「ハイ，セーブ！」と一発で処理できる便利な関数です。

YET再び

6月号付録ディスクのYET.Xはトップ10のスコアをファイルに残します。もともとオマケ的な要素が強かったので暗号化も行わず，単純に名前とスコアを記録するようになっています。DUMP.Xで覗くとその構造がよくわかるでしょう。作成当時には最高得点は3万点台が限界だろうと思い，このあたりなら十分自分の名前を残すことができるという自負から，同じ人物の得点は最高点のみを残すなどという細工を行わなかったのでした。

あぁそれなのに，それなのに。編集室ではいつしかトップ10すべてが4万点台になってしまったのです。しかもたった2人の人物によって！　結局私はやってもやってもスコアを残すことができず，「これはなんとかしなければ」という使命感のもと，スコア調整プログラムを作ることにしました。

このスコア調整プログラムは次の2つの機能を持っています。

1)　同じ人物のスコアは最高得点のみを残す
2)　2つのスコアファイルを融合する

1)は1人の人物がスコアを独占し，ほかの人が名前を登録する栄誉にあずかれないという事態を打破するために用意しました。2)は自宅でさんざんやって出した高得点をクラブのX68000に移し，友達に尊敬されるためです。

●YETSCOのファイル構造

YETのスコアファイルであるYETSCOは整数型の配列をfwrite関数でファイルに書き出しただけの非常に簡単な構造をしています。1人分のデータは，

1. 名前の1文字目のASCIIコード
2. 名前の2文字目のASCIIコード

　　⋮

6. 名前の6文字目のASCIIコード
7. スコア

という形式で7つの整数型データに変換され，これが10人分続いたのがスコアファイルなのです。例をお見せしましょう。「DAISKE　32000」というスコアをこの方法で変換すると，

68 65 73 83 75 69 32000

となります。

●まずはスコアファイルの読み込みから

ではまず，スコアファイルの読み込みです。スコアファイルはfwriteで書き出したファイルですので，読み込みはfreadで行いましょう。1人のデータが整数7個分ですから，10人のデータは整数70個分になります。

```
int scoFile(70)
```

でデータを読み込む1次元配列を作成し，

```
int file
file=fopen("yetsco","r")
fread(scoFile, 70, file)
fclose(file)
```

でデータの読み込みは終了です。freadは読み込む配列名と，読み込むデータの個数，そしてファイル番号を引数にとります。

●名前とスコアを取り出す

データをいったん読み込んでしまえば，あとは普段のプログラミングと変わりありません。これまで初期値を与えた配列を使うプログラムをいくつか作ってきましたが，初期値を与える代わりにファイルから読み込んだだけだと考えてもいいでしょう。

いま，(ファイルから読み込んで) 初期値を与えた配列scoFileがあります。これはASCIIコードとスコアをごちゃまぜにして登録してある配列です。このままでは扱いづらいので，プレイヤーの名前を入れた配列と，スコアを入れた配列に分けることにします。

```
str player(9)
int score(9)
```

の２つの配列を用意し，これにscoFileのデータを取り出してセットします。

プレイヤーの名前は必ず６文字分とってscoFileに収めてあり，そのあとにスコアがセットしてありますから，

```
for i=0 to 9
  for j=0 to 5
    player(i)=player(i)
      +chr$(scoFile(i×7+j))
  next
  score(i)=scoFile(i×7+6)
next
```

として２重ループを作ればplayerとscoreの２つの配列にデータをセットすることができます。６つの名前データと１つのスコアデータが１組になっていますから，i×7番目から６つのデータを取り出しそれを文字列に変更してplayer配列のｉ番目に，その次のデータを取り出してscore配列のｉ番目にセットしているのが上のプログラムです。

ではここでプログラムを見ていただきましょう。リスト２です。10行ではスコアファイル名をユーザが設定できるように変数として宣言しています。20行はいま説明したデータ読み込み用配列，そして30，40行がplayer配列とscore配列です。ここでは３つ宣言してありますね。これは，このあと同じ名前の削除を行うのに，ひとつの配列の中でやりくりするのは面倒なためです。加工後のデータは別の配列に入れることにしました。

上で説明したファイル読み込みおよびplayer，score配列へのセットを行っているのは1160行のreadSco関数です。ここでは引数ｎの値によって，player1，score1にセットするのか，player2，score2にセットするのかを振り分けています。

●同一人物の削除

同一人物を削除するには，同じ名前を飛ばしてスコア配列を詰めていけばOKです。readSco(1)でplayer1，score1配列にデータを読み込み，player1配列を上から順に見ていって，初めて登場する名前なら名前とスコアをplayer，score配列へ移します。

問題は初めて登場する名前かどうかを判定する方法です。player配列を順に調べてもいいのですが，ここではinstr関数を使うことにしました。instr関数は，文字列が特定の文字列を含んでいるかどうかを判定する関数です。player1配列からplayer配列へ移した名前を文字型変数chkStrに順次代入していくことにすれば，ある名前をplayer配列に移したかどうかはchkStrを調べるだけですみます。

ここで気をつけなければならないのは，文字列を単純に追加してはいけないということです。スコアのトップがdai，２番目がdanだったとします。単純に追加するとchkStrは，

daidan

となりますね。スコアの３番目がidaだとすると，idaはすでにchkStrに入っていることになってしまうため，player配列へ移されません。

このような事態を避けるため，名前の前後を決して名前に使われない文字で区切る必要があります。決して名前に使われない文字を仮に‘．’だとすると，chkStrは，

.dai.dan.

となり，“.ida.”はこの中に含まれないのでうまくいきます。

これらの処理を行っているのが320行から始まるunify関数です。ここでは区切り文字としてchr$(1)を使っています。

●整頓後のスコアの保存

スコアの保存はスコアの読み込みと逆の手順で行います。整頓が終わったスコアはplayer，score配列に収められていますから，これら２つの配列からscoFile配列へデータを移し，それをfwrite関数で一気に書き出せばOKです。これは1570行のsaveSco関数が行っています。

●２つのスコアファイルを融合する

player2，score2配列が用意してあるのは，この機能を実現するためです。１つ目のファイルをplayer1，score1配列に，２つ目のファイルをplayer2，score2配列にセットし，これら２つの配列から点数の大きいもの順にplayer，score配列へと移していくと融合が完成します。具体的には２つの配列の添字用に２つの変数（rank1，rank2）を用意し，

```
if score1(rank1)>=score2(rank2) then {
  配列 player1，score1をplayer，scoreへ
  rank1=rank1+1
} else {
  配列 player2，score2をplayer，scoreへ
  rank2=rank2+1
}
```

とします。添字変数はデータを移したときだけ大きくなり，次のスコアがもう一方のスコアと比較されることになります。

この処理を行っているのが730行から始まるmerge関数です。画面表示処理が間に入っているので若干わかりづらいかもしれませんが，やっていることは上で説明したことだけです。

●プログラムの拡張について

さて毎度のことながら，プログラムには必要最小限の機能しか盛り込んでありません。エラー処理はまったくやっていませんし（X-BASICで実行する

なら，致命的なエラーはBASICが出してくれる)，処理を途中でやめたくなった場合のことも考慮してありません。整頓終了後に画面に表示される結果が気に入らない場合は，ファイル名入力のプロンプトが表示されているときにブレイクしてください。

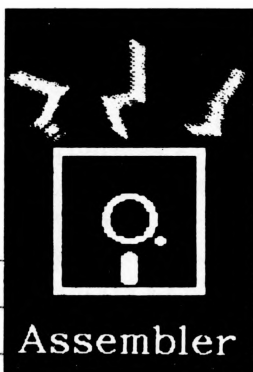
まず最初に皆さんに取り組んでもらいたい拡張は，2つのファイルを融合するときに同一人物を削除する機能を付加することです。unify関数が参考になるかと思います。

yetscoファイルを読み込み変数にセットするこのプログラムを使えば，簡単にスコアを変更することができてしまいます。player，score配列の中身を適当にいじってsaveSco関数を呼び出すだけでいいのですから5万，10万点のスコアなんて楽勝です。そんなスコアを見せびらかして喜ぶような悲しい遊びはやらないでくださいね。

来月は「ちょっと高度なファイル処理」と称してデータベースもどきをお送りする予定です。

リスト2　YETのスコア管理ぷろぐらむ

```
10 str scoName                      /* スコアファイル名用
20 int scoFile(70)                  /* データ読み込み用1次元配列
30 str player(9), player1(9), player2(9)
40 int score(9), score1(9), score2(9)
50 int conFlag = 1
60 str selection
70 /*
80 while conFlag
90   cls
100   print "Score Manager"
110   print
120   print "1)  同一人物削除"
130   print "2)  ファイル融合"
140   print "3)  終了"
150   print
160   print "処理する番号:";
170   selection = inkey$
180   switch selection
190     case "1"
200       unify()
210       break
220     case "2"
230       merge()
240       break
250     case "3"
260       conFlag = 0
270       break
280   endswitch
290 endwhile
300 end
310 /*
320 func unify()
330   str chkStr[80]
340   int i, rank=0
350   /*
360   /* スコアファイル読み込み
370   /*
380   locate 0, 10
390   input "スコアファイル名:",scoName
400   readSco( 1 )
410   cls
420   for i=0 to 9
430     print player1(i), score1(i)
440   next
450   print
460   /*
470   /* 同一人物削除
480   /*
490   print "   変換"
500   print
510   chkStr = chr$(1)
520   for i=0 to 9
530     player(i) = ""
540     score(i) = 0
550     if instr( 1, chkStr, chr$(1)+player1(i)+chr$(1) ) = 0 then
560       chkStr = chkStr + player1(i) + chr$(1)
570       player(rank) = player1(i)
580       score(rank) = score1(i)
590       rank=rank+1
600     }
610   next
620   for i=0 to 9
630     print player(i), score(i)
640   next
650   print
660   /*
670   /* スコアファイル保存
680   /*
690   input "セーブします。ファイル名:", scoName
700   saveSco()
710 endfunc
720 /*
730 func merge()
740   int rank, rank1, rank2
750   /*
760   /* 2つのスコアファイル読み込み
770   /*
780   locate 0, 10
790   input "スコアファイル名1:", scoName
800   readSco( 1 )
810   input "スコアファイル名2:", scoName
820   readSco( 2 )
830   cls
840   for i=0 to 9
850     print player1(i), score1(i),,player2(i), score2(i)
860   next
870   print
880   /*
890   /* 2つのスコアを1つにまとめる
900   /*
910   print "   変換"
920   print
930   rank1=0 : rank2=0
940   for rank=0 to 9
950     if score1( rank1 ) >= score2( rank2 ) then {
960       player( rank ) = player1( rank1 )
970       score( rank ) = score1( rank1 )
980       rank1 = rank1 + 1
990     } else {
1000       player( rank ) = player2( rank2 )
1010       score( rank ) = score2( rank2 )
1020       rank2 = rank2 + 1
1030     }
1040   next
1050   for i=0 to 9
1060     print player(i), score(i)
1070   next
1080   print
1090   /*
1100   /* スコアファイル保存
1110   /*
1120   input "セーブします。ファイル名:", scoName
1130   saveSco()
1140 endfunc
1150 /*
1160 func readSco( n )
1170   int file
1180   int i
1190   str ch
1200   /*
1210   /* 対象とするスコア配列にデータ読み込み
1220   /*
1230   file = fopen( scoName, "r" )
1240   fread( scoFile, 70, file )
1250   fclose( file )
1260   /*
1270   /* 対象とする名前配列をクリア
1280   /*
1290   for i=0 to 9
1300     if n = 1 then {
1310       player1(i) = ""
1320     } else {
1330       player2(i) = ""
1340     }
1350   next
1360   /*
1370   /* 名前配列に名前を
1380   /*   スコア配列にスコアをセット
1390   /*
1400   for i=0 to 9
1410     for j=0 to 5
1420       ch = chr$( scoFile( i*7 + j ))
1430       if n = 1 then {
1440         player1(i) = player1(i) + ch
1450       } else {
1460         player2(i) = player2(i) + ch
1470       }
1480     next
1490     if n = 1 then {
1500       score1(i) = scoFile( i*7 + 6 )
1510     } else {
1520       score2(i) = scoFile( i*7 + 6 )
1530     }
1540   next
1550 endfunc
1560 /*
1570 func saveSco()
1580   int file
1590   int i
1600   /*
1610   /* 名前配列から名前を
1620   /* スコア配列からスコアを取り出し
1630   /* scoFileにセット
1640   /*
1650   for i=0 to 9
1660     for j=0 to 5
1670       scoFile( i*7 + j ) = asc( mid$( player(i), j+1, 1 ))
1680     next
1690     scoFile( i*7 + 6 ) = score( i )
1700   next
1710   /*
1720   /* scoFileを書き出す
1730   /*
1740   file = fopen( scoName, "c" )
1750   fwrite( scoFile, 70, file )
1760   fclose( file )
1770 endfunc
```



マウスwithグラフィック

Murata Toshiyuki 村田 敏幸

X68000用のプログラムを作成するというのなら，やはり，マウスも基本として押さえておきたいところ。マウス制御のためのさまざまな機能がIOCSとして用意されていますからこれを利用するのが正攻法です。簡単なお絵かきツールで実践してみましょう。

最初に，前回の記事中にポカがあったので訂正しておく。アドレスレジスタにaddqやsubqで小さな定数を加減算するときにワードサイズを指定したほうが速いと書いたが，大嘘なので忘れてほしい。実際には，ロングワードでもワードでも実行速度は変わらない。また，最後のASX.Sの中で使っているインクルードファイルが抜けていた。リスト0にそのFILES.Hを示す。ひと月休んで訂正が遅れたことと合わせてお詫びする。痛惜の念に堪えない，ぐらいのことはいうべきなのかもしれないが，この言葉はいつかとんでもない大バグを出したときのためにとっておこうと思う。

＊

さて，今回は地味ながらX68000らしいところでマウスを取り上げ，最後はこれにパラパラとグラフィックを振りかけてこぢんまりとまとめてみたい。あくまでマウスがメインであり，グラフィックまわりについてはあまり詳しく触れないことをあらかじめ断っておく。

IOCSコールを使う

X68000ではROMにIOCS（Input/Output Control System）の形でさまざまな機能の制御ルーチンが用意されており，マウスもこのIOCSを呼び出すことによってほとんどX-BASICと同じ感覚で手軽に利用することができる。もうご存じだとは思うが，一応，IOCSの概要と呼び出し手順を押さえておこう。

X68000のIOCSはテキスト画面への文字表示，キー入力に始まって，プリンタ出力，RS-232Cによる入出力，フロッピーディスク/ハードディスクの物理的な読み書き，マウスの制御，グラフィック描画，スプライト，AD PCM，カレンダー時計などの周辺LSIの制御にいたるまで，X68000の（ほとんど）すべての機能を網羅している。位置づけとしてはシステム中もっともハードに近い部分を担当しX68000上のプログラムを底辺からささえる低レベルI/Oルーチン集[1]であり，OSであるHuman68kもIOCSに乗っかった形で作られている。

これにより，ユーザープログラムがHuman68kに入出力を要求すると，Human68kは必要に応じてIOCSを呼び出し，最終的にIOCSがハードに働きかけて物理的な入出力を行う[2]。結果は逆のルートを伝って返される。このHuman68kとIOCSとの上下関係（というか依存関係というか階層構造というか）は心に留めておいてもらいたい。

IOCSを呼び出す手順はいたって簡単で，d0.lにIOCSコール番号を入れてtrap #15という命令を実行するだけだ。パラメータがあるときはd1以下のデータレジスタやa1以下のアドレスレジスタ（a0はIOCSコール呼び出しには使われない）に入れて渡す。たとえば，IOCSコール番号20Hに割り当てられている1文字表示機能を使うときには，

```
        move.w    #'A',d1
        moveq.l   #$20,d0
        trap      #15
```

IOCSコール番号21Hの文字列表示機能を使うのなら，

```
        lea.l     mes,a1
        moveq.l   #$21,d0
        trap      #15
              :
```

1）実際にはハードがらみ以外にも，シフトJIS漢字コード←→JIS漢字コードの相互変換とか，ユーザーモードからスーパーバイザ空間にあるメモリを読み書きするといったユーティリティ的なものもIOCSには用意されている。

2）論理的には，OSの低レベルI/Oはデバイスドライバが担当することになっているわけだが，現実にはHuman68kのデバイスドライバはさらに下位の存在であるIOCSを下請けに使っている場合が多い。

trap命令

trapは端的にいうと故意に例外を発生させる命令だ。trapにはtrap #0～#15の16個があり，順に例外ベクタ番号20H～2FH，例外ベクタアドレスでいうと0080H以降の16ロングワードが割り当てられている。

trap命令が実行されると68000はスーパーバイザモードに移行し，命令が実行された時点でのpcとsrの値をスーパーバイザスタックに積む。そののち，該当する例外ベクタの内容を参照し，指定されたアドレスから例外処理を実行する。DOSコールの呼び出しに利用されている未実装命令の実行による例外とは異なり，trap命令による例外処理開始時にスタックに積まれるpcは命令が置かれた直後のアドレスを指しており，小細工をしなくとも例外処理の最後でrteを実行すればtrap命令のすぐうしろからプログラムの実行を再開できる。

感覚としては“スーパーバイザモードへの移行を伴うサブルーチンコール命令”といったようなもので，その性質上，システムコールを呼び出すのによく用いられている（そのようにシステムが設計される）。本文でも触れたようにX68000ではIOCSの呼び出しにtrap #15を使っている。

このほかX68000＋Human68kではtrap #8～#14を内部的に使用している。ふつうのプログラムを作るうえでは知らなくてもすむのだが，興味のある人は「プログラマーズマニュアル」の3.2節末にある参考資料を見てみるとよいだろう。

```
mes:    .dc.b   '文字列',0
```

という具合だ。ソースプログラム中にIOCSコール番号を生のまま埋め込むのがいやであれば，Human68kのDOSコールの場合のように，インクルードファイルをひとつ作成してその中で，

```
__B__KEYINP     equu    $00
        ⋮
__B__PUTC       equ     $20
__B__PRINT      equ     $21
        ⋮
```

のようにずらずらとIOCSコール番号をシンボル定義しておけばよい。幸いなことにXCにはこのインクルードファイルがIOCSCALL.MACの名前であらかじめ用意されている。また，IOCSCALL.MAC内では，

```
IOCS    macro   callno
        moveq.l #callno,d0
        trap    #15
        endm
```

というマクロが定義されていて，このマクロとシンボルを利用すると上の例は，

```
move.w  #'A',d1
IOCS    __B__PUTC
```

とか，

```
lea.l   mes,a1
IOCS    __B__PRINT
```

のようにすっきり書けるようになる。今後この連載でIOCSコールを利用するときにはIOCSCALL.MACをインクルードし，このスタイルで記述する（編集部注：本誌6月号の付録ディスクにも収録させていただいたので利用してください）。

実際にIOCSコールを使ったプログラムの一例をリスト1に示す。こんな機会でもなければ誌面に載ることもないようなちっぽけなプログラムLEDOFF.Xだ。実行するとすべてのLEDキーをOFF状態にする。AUTOEXEC.BATに潜り込ませるか，Human68k Ver.2ならCONFIG.SYSのPROGRAM=～行に記述するかして起動時に1回走らせるのが正しい使い方だ。起動直後に“ぢ”とか“ヂ”と打ち込んで“コマンドまたはファイル名が違います”攻撃を受けたことがある人ならLEDOFF.Xの有用さに気づいてもらえると思う。

見てのとおりプログラムはLEDキーの状態を操作するIOCSコールLEDMODをループの中から発行するだけという単純さだ。LEDMODは2つのパラメータを採り，d1.lでLEDキーの番号（0～6），d2.bでONにする（1）かOFFにする（0）かを指定する。リスト1ではループ内でd2を0に固定したままd1を順に変化させてすべてのLEDキーをOFFにしていっている。このことからも察しがつくと思うが，IOCSコールでは基本的にd0以外のレジスタの値は保存される（例外はある）。d0だけはIOCSコールの終了ステータスないしは適当な戻り値を返すのに使われる。ちなみに，LEDMODはパラメータの値が範囲外でLEDの設定できなかった場合は－1を，うまく設定ができたときは0をd0.lに返す。

リスト0　FILS.H

```
*         nameck,files,nfiles用オフセット定義

          .offset 0
*
DRIVE:    .ds.b   2          *ドライブ名   'A:'
PATH:     .ds.b   64+1       *パス名       '¥BIN¥',0
NAME:     .ds.b   18+1       *ファイル名   'ATTRIB',0
EXT:      .ds.b   1+3+1      *拡張子       '.X',0
          .even
NAMBUFSIZ:
*
          .offset 0
*
FORSYS:   .ds.b   21         *システムが使用
FATR:     .ds.b   1          *ファイル属性
FTIME:    .ds.w   1          *ファイル最終更新時刻
FDATE:    .ds.w   1          *ファイル最終更新日
FLEN:     .ds.l   1          *ファイル長
PACKEDNAME:                  *ファイル名
          .ds.b   18+1+3+1
          .even
FILBUFSIZ:
*
          .text
```

リスト1　LEDOFF.S

```
*         全てのLEDキーをOFFにする
*
          .include        iocscall.mac
          .include        doscall.mac
*
ent:
          moveq.l #0,d2     *OFF
          moveq.l #7-1,d1   *LEDキー番号
loop:     IOCS    _LEDMOD   *設定
          dbra    d1,loop   *繰り返す

          DOS     _EXIT     *終了

          .end    ent
```

そしてマウスへ

とんとんとマウスの話に進む。マウス関連のIOCSコールはコール番号70_H～$7D_H$にまとめられており，『プログラマーズマニュアル』を見てもらえればわかるように，X-BASICのマウス操作関数と似たような機能を持ったものがずらっと並んでいる。

X-BASICでマウスを扱ったことがあれば，これらを使いこなすのもわけはない。さっさとサンプルにいってしまってかまわないだろう。リスト2のMSTEST.Sは画面にメニューをひとつ表示し（実際には“終了”という文字列を左上隅に書くだけ），このメニューの上で左ボタンが押されたら，それに応じた処理をする（終了する）プログラムだ。

11～14行はマウスを使うときには枕詞のように現れる初期化・使用準備の決まりきった手順だ。最初のMS__INITによりマウスカーソルの表示はOFFになり，カーソルパターンは標準の矢印型に，カーソル座標は（0,0）に，カーソルの移動範囲は表示画面の大きさと一致するように初期化される。つづくMS__CURONでマウスカーソルを表示状態にし，SKEY__MODでマウスの右ボタンに割り当てられているソフトウェアキーボードとマウスカーソルの表示/非表示切り換え機能を殺して初期化は完了だ。この3つのIOCSコールの組み合わせは，X-BASICの

mouse（0）
mouse（1）

mouse (4)

にほぼ対応している。

16行からメイン処理が始まる。まず，左ボタンが押されるまで待つ（16～18行）。ボタンの状態を得るにはIOCSコールMS_GETDTを利用する。このIOCSコールはX-BASICのmsstat（）に相当し，d0.lの上位ワードにマウスカーソルの相対的な移動量を，下位ワードに左右のボタンの状態を返す。相対的なカーソル移動量のほうはあまり利用されることはないはずだからここでは触れない。

ボタンの状態は第0～7ビットに右ボタン，第8～15ビットに左ボタンのON/OFF状態が返り，ボタンが押されているときは8ビットとも1（FF_H），押されていなければ8ビットとも0（00_H）になる。右ボタンが押されているかどうかチェックしたければ，

```
IOCS    _MS_GETDT
tst.b   d0
beq     押されていない
押されている～
```

のようにtst.b後のZビットで処理を振り分ければよいのは明らかだろう。左ボタンの場合は，

```
IOCS    _MS_GETDT
tst.w   d0
bpl     押されていない
押されている～
```

という手が使える。どうせ8ビットとも同じ値をとるのだから，第15ビットだけを調べればすむわけだ。

MS_GETDTで左ボタンの押し下げが検出されたら，すかさずMS_CURGTでマウスカーソルの画面上での現在位置を得る（21行）。MS_CURGTはX-BASICのmspos（）関数にあたり，d0.lの上位ワードにマウスカーソルのX座標，下位ワードにY座標を返す。得られた座標がメニュー上にあるかどうかを調べているのが25～28行，やっているのは単純な座標の比較だ。

最後に31行以下が忘れてはならない後始末の処理だ。MS_INITでマウスを再初期化して（マウスカーソルを消し），SKEY_MODでさっき殺したソフトウェアキーボードを使用可能状態に戻している。

マウスについてはだいたいがリスト2の応用で片がつく。あと，ダブルクリックの判定方法ぐらいは知っていたほうがいいかもしれない。そこでリスト3。リスト2の30行以下と差し換えて使う。ダブルクリックの判定といってもやるべきことは泥臭いといっていいほど直接的だ。ボタンが押されたことがわかったら，

1) 一定時間以内に離されるかどうか

2) 一定時間以内にまた押されるかどうか

というチェックを続けて行い，両方に通ったらダブルクリックされたと判断する。これには，IOCSコールのMS_OFTM，MS_ONTMを利用する。d1.wで左右のボタンのどちらか（0なら左，−1なら右），d2.wで待ち時間を指定し（とくに0のときは無限と見なされる），指定時間内にボタンが離されたり（MS_OFTM）押されたり（MS_ONTM）したら，それまでの経過時間をd0.wに返す。ただし，ドラッグされた場合（ボタンの状態が変化しないうちにマウスカーソルが動いた場合）にはd0.w＝0で即戻ってくる。また，待ち時間を越えた場合は$FFFF_H$が返る。待ち時間の単位はなにやらいい加減らしく（ループ回数で計時しているのかな），だいたい40が0.1秒前後に相当する。リスト3では待ち時間を0.2秒程度にするために80を指定してある。

リスト2 MSTEST.S

```
 1:         .include        iocscall.mac
 2:         .include        doscall.mac
 3:         .include        const.h
 4: *
 5: ent:
 6:         lea.l   mysp(pc),sp
 7:
 8:         lea.l   menu(pc),a1        *メニューを描く
 9:         IOCS    _B_PRINT           *
10:
11:         IOCS    _MS_INIT           *マウス初期化
12:         IOCS    _MS_CURON          *マウスカーソル表示
13:         moveq.l #0,d1              *ソフトウェアキーボード
14:         IOCS    _SKEY_MOD          * 表示禁止
15:
16: loop:   IOCS    _MS_GETDT          *ボタンの状態を得る
17:         tst.w   d0                 *左ボタンは押されているか?
18:         bpl     loop               * 押されていなかった
19:
20:                         *左ボタンが押された
21:         IOCS    _MS_CURGT          *マウスカーソル座標を得る
22:         move.w  d0,d1              *d1.w = Y座標
23:         swap.w  d0                 *d0.w = X座標
24:
25:         cmpi.w  #32,d0             *X座標のチェック
26:         bcc     loop               * 範囲外
27:         cmpi.w  #16,d1             *Y座標のチェック
28:         bcc     loop               * 範囲外
29:
30:                         *終了メニュー上だった
31:         IOCS    _MS_INIT           *マウス再初期化
32:         moveq.l #-1,d1             *ソフトウェアキーボード
33:         IOCS    _SKEY_MOD          * 表示許可
34:
35:         DOS     _EXIT              *終了
36: *
37:         .data
38:         .even
39: *
40: menu:   .dc.b   26,'終了',CR,LF,0
41: *
42:         .stack
43:         .even
44: *
45: mystack:
46:         .ds.l   256
47: mysp:
48:         .end    ent
```

リスト3 MSTEST2.S

```
30:                         *終了メニュー上だった
31:         moveq.l #0,d1              *左ボタン
32:         moveq.l #80,d2             *待ち時間（約0.2秒）
33:         IOCS    _MS_OFFTM          *離されるまで待つ
34:         tst.w   d0                 *0以下なら
35:         ble     loop               * はじく
36:
37:         IOCS    _MS_ONTM           *押されるまで待つ
38:         tst.w   d0                 *0以下なら
39:         ble     loop               * はじく
40:
41:                         *ダブルクリックされた
42:         IOCS    _MS_INIT           *マウス再初期化
43:         moveq.l #-1,d1             *ソフトウェアキーボード
44:         IOCS    _SKEY_MOD          * 表示許可
45:
46:         DOS     _EXIT              *終了
47: *
48:         .data
49:         .even
50: *
51: menu:   .dc.b   26,'終了',CR,LF,0
52: *
53:         .stack
54:         .even
55: *
56: mystack:
57:         .ds.l   256
58: mysp:
59:         .end    ent
```

お絵かきツールへの応用

最後に応用プログラムとして，簡単なお絵かきツール（グラフィックエディタなんて呼べるほどの代物ではない）を作って今月はおしまいにする。当初はマウスボタンが押されたらその位置に点を打つだけのプログラムにしようと思っていたが，これだとあまりに単純すぎて面白みに欠けるので，IOCSコールで実現できる範囲で多少彩りを添えてみた。

1)　色の選択は右ボタンを押すことでポップアップするウィンドウで選べるようにする

2)　同じウィンドウ上にはペンパターンのメニューも並べ，複数の中からペンのパターンを選べるようにする（パターンは最大16×16ドット）

一見複雑な処理が要求されそうだが，X68000のハードの機能とIOCSのおかげで，どちらも簡単に実現できる。まず，ウィンドウをポップアップする処理だが，256色２画面の画面モードを使用して，１画面をウィンドウ用，残りを描画用と使い分けることで逃げた。ウィンドウはあらかじめ全部描いておき，X-BASICのvpage関数，home関数に相当するIOCSコールVPAGEとHOMEで表示のON/OFF，表示位置の変更を行う。２点目のペンパターンについては，“外字をSYMBOLで表示する”という手を使った。ペンのパターンを外字に登録しておき，PSETで点を打つ代わりにSYMBOLで描くわけだ。どちらもかなり安直だが，彩りとしての役目は果たしてくれる。

グラフィック関係のIOCSについては約束どおり特に解説しないから『プログラマーズマニュアル』を参照してもらいたい。一応リスト４にLINEのサンプルを示しておく。COMMAND.X上からグラフィック画面に直線を描画するプログラムだ。７行のパラメータの個数と，23行のIOCSコール番号を変更すればBOXやFILL，CIRCLEにも対応できるので気が向いたら試してみてほしい。あまり使い道のないプログラムだが，派手なバッチファイルを作りたいときなんかには利用できるだろう。

なお，コマンドラインで指定された数字（の文字列）を数値に変換するのにリスト５中のサブルーチンatoiを利用しているので，実行ファイル作成時にはこれも忘れずにリンクすること。このatoiは今後も使うことがあるかもしれない（変に凝ってしまったのであまりよいできではないが）。また，LINE.Xはグラフィック画面の初期化を行わないので，使用時にはSCREENコマンドであらかじめグラフィック画面を使用可能に設定しておく必要がある。

atoiについて１点だけ補足しておく。５～８行ではＣフラグを反転（０⟷１）するマクロCCFを定義している。その実体は，

```
eori.w #1,ccr
```

というオペランドにccrが登場するという見慣れない命令だ。この命令は任意のフラグを反転するのに使う。排他的論理和の意味と，ccrの構造を思い出してもらいたい。同様の命令としては，

```
andi.w  #n,ccr
ori.w   #n,ccr
```

があり，それぞれ，ccrレジスタ中の任意のフラグをリセットしたりセットしたりするのに用いられる。

STAMP.Sの解説

では，手抜きいっぱいのお絵かきプログラム，リスト６のSTAMP.Sを見てもらおう。比較的読みやすく書けたと思うので，これまでの話のまとめのつもりで読んでみてもらいたい。各ルーチンごとにポイントとなる部分を拾って軽く解説しておく。

●エントリ～終了（62行～）

Interruptスイッチなどによってプログラムの実行が中断された場合に後始末をせずに親プロセスに帰るのがいやだったので，67～72行で前回のASX.Xとまったく同じ手順で中断時の戻りアドレスを77行のラベルbreakの位置に設定している。

break以降では諸々の後始末をするサブルーチンを呼び出してから，キーバッファをクリアし，exitで実行終了する。マウスしか使わないプログラムでキーバッファを気にしているのが変に見えるかもしれないが，“マウスしか使わないからこそ”この処理が必要なのだ。これを怠ると，プログラム走行中に誤って押されたキーがプログラム終了後にまとめて吐き出されることになる。

●初期化ルーチン（275行～）

278～290行でDOSコールconctrlによって画面モードを横512×縦512ドット，256色モードに切り換えたうえで，邪魔なファンクションキー行とカーソルを消している。画面モードとファンクションキー行についてはあとで元に戻せるように（374行以下の後始末ルーチン参照）現在の状況をワークエリアにしまっておく。それが作法というものだ。あと，このサブルーチンでは頭でlinkし，リターンする直前でunlkすることによってDOSコール呼び出し時のスタック補正を省略するという姑息なテクニックが使われている。あまり褒められたことではないが，一度やって見せたかった。

293，294行は下位のサブルーチンを呼び出して，ペンパターンとして利用する外字の定義を行っている。ここでも，あとで元に戻せるように現在の外字の定義を取得・待避しておくのを忘れない。定義する外字のフォントパターンは436行以下に用意してあり，16ワードが１文字分のデータにあたる。

頭に縦横のドット数がつけてあるのはほかとの兼ね合いで，実際には使っていない。フォントパターンは438～453行の最初の１個だけは見やすく２進数で表記してみた（２個目以降はスペースの都合で詰めて16進数で表記してある）。これを見ればフォントパターンの形式・作り方は一目瞭然だろう。

●メニューウィンドウの初期化（309行～）

前述のとおり，メニューはあらかじめ全部描いておく。描画に必要なデータはデータセクションに用意しておき，これを次々にIOCSコールに渡している。

●メイン処理（88行〜）

多少冗長な作りになっているが，マウスのボタンの状態をチェックし，ボタンが押されていたらその位置に応じてそれなりの処理を行うというパターンの組み合わせであり，リスト２と基本的には大差ない。左ボタンが押された場合は，まずメニューウィンドウ上かどうかを調べ，ウィンドウ外（もしくはウィンドウが非表示状態）であれば197行に飛んでSYMBOLで現在設定されているペンパターン（に対応する外字１文字）を描く。ウィンドウ上だった場合は，マウスカーソル座標から，

1)　ペン選択メニュー上
2)　色選択メニュー上
3)　終了メニュー上
4)　いずれでもないウィンドウの外枠

を識別し，対応する処理を行う。1)，2) の場合はさらにメニュー上のどの部分かの判定が加わることになる。また，ウィンドウの外枠で左ボタンが押された場合はウィンドウをドラッグするようにしてみた。本来ならマウスの動きに連動してリアルタイムでウィンドウの位置を変更することもできたのだが，もっと単純に，ボタンが離された位置へいきなりウィンドウを移動するようになっている。ここは読者に手を入れてもらいたい部分のひとつだ。

左ボタンの処理に比べれば，207行以下の右ボタンによるメニューウィンドウのON/OFF切り換え処理はシンプルだ。現在メニューが表示中かどうかを覚えておくワークmenuflagを調べて（209行），もしメニューがすでに表示中であれば212行以下でVPAGEによりメニューが描かれているページを非表示にする。メニューが表示されていなければ221行以下で現在のマウスカーソルの位置にメニューを表示する。

なお，222行でmenuflagをセットするのに使っているst.bは，任意の１バイトをFF_Hにする命令だ（オペランドサイズはバイト固定）。正確にはstの一般形はsXX（sはSetの略）であり，XXの部分には条件分岐命令同様の条件が入る。sXXは命令実行の時点でこの条件が成り立っていればオペランドをFF_Hにし，条件が成り立っていなければ00_Hにする命令で，stはこの条件が“t（always True：常に真）”になった形だ。条件が常に成り立つわけだから，オペランドを00_Hにすることはありえない。逆にsfという命令は条件が“f（always False：常に偽）”であり，任意の１バイトを00_Hにするのに使える。趣味の問題だが，人によってはclr.bの代わりに使うこともある。

＊

『プログラマーズマニュアル』をパラパラと眺めてみると，それ単体でプログラムとして成り立つようなIOCSコールがいくつか見つかると思う。例を挙げるなら，コール番号$7F_H$のONTIME（本体を立ち上げてからの時間を100分の１秒単位で返す）とか$8E_H$のBOOTINF（前面の電源スイッチにより起動されたのか，タイマにより起動されたのか，また，どのデバイスから起動されたのかといったブート情報を返す）なんかは，IOCSコールからの戻り値を表示するだけでもそれなりに役にたつ（ことがあるかもしれない）プログラムになる。この類のプログラムはあって困るものでもなし，暇を見つけて作っておくとよいだろう。

来月は，グラフィックをもう少し本格的に取り上げる予定でいる。

リスト4　LINE.S

```
         .include        doscall.mac
         .include        iocscall.mac
         .include        const.h
*
         .xref   atoi                   *外部参照
*
PARCNT   equ     6              *IOCSに渡すパラメータの個数
*
         .text
         .even
*
ent:
         lea.l   mysp(pc),sp            *spを初期化する

         bsr     getpar                 *パラメータを取得する

         moveq.l #-1,d1                 *グラフィック画面は
         IOCS    _APAGE                 *  初期化されているか?
         tst.b   d0                     *
         bmi     error                  *未初期化ならエラー終了

         lea.l   giocspar(pc),a1 *直線描画
         IOCS    _LINE                  *
         tst.b   d0                     *エラー?
         bmi     usage                  *  パラメータの値が変

         DOS     _EXIT                  *正常終了

*
*        PARCNT個の数値をバッファにセットする
*
getpar:
         tst.b   (a2)+                  *空文字列なら
         beq     usage                  *  使用法を表示して終了

         lea.l   giocspar(pc),a1 *a1=パラメータ格納バッファ
         moveq.l #PARCNT-1,d1           *d1=ループカウンタ
getpr0:  move.l  a2,-(sp)               *文字列→数値変換
         bsr     atoi                   *
         movea.l (sp)+,a2               *a2=続く文字列
         bmi     usage                  *うまく変換できなかった
         move.w  d0,(a1)+               *パラメータを格納
         dbra    d1,getpr0              *PARCNT回繰り返す

         rts

*
*        使用法の表示&エラー終了
*
usage:
         move.w  #STDERR,-(sp)          *標準エラー出力へ
         pea.l   usgmes(pc)             *  メッセージを
         DOS     _FPUTS                 *  出力する
         addq.w  #6,sp                  *
         *
error:   move.w  #1,-(sp)               *終了コード1を持って
         DOS     _EXIT2                 *  エラー終了

*
*        データ&ワーク
*
         .data
         .even
*
usgmes:  .dc.b   '機　能:グラフィック画面に直線を描きます'
         .dc.b   CR,LF
         .dc.b   '使用法:LINE X0 Y0 X1 Y1'
         .dc.b           ' パレットコード ラインスタイル'
         .dc.b   CR,LF,0
*
         .bss
         .even
*
giocspar:
         .ds.w   PARCNT                 *パラメータバッファ
*
         .stack
         .even
*
mystack:
         .ds.l   256                    *スタック領域
mysp:
         .end    ent
```

リスト5　ATOI.S

```
         .include      const.h
*
         .xdef   atoi
*
CCF      macro                           *Cビットを反転するマクロ
         eor.w   #%00001,ccr             *
*                  XNZVC                 *
         endm                            *
*
TOUPPER macro    dreg                    *英小文字→大文字変換マクロ
         local   skip                    *
         cmpi.b  #'a',dreg               *
         bcs     skip                    *
         cmpi.b  #'z'+1,dreg             *
         bcc     skip                    *
         subi.b  #'a'-'A',dreg           *
skip:
         endm
*
         .text
         .even
*
*atoi(str)
*機　能:数値を表す文字列を16ビット符号付整数に変換する
*戻り値:d0.w    =  変換された値
*       (sp).l  =  続く文字列へのポインタ
*       N       =  1文字も変換できなかった場合に1
*メ　モ:文字列先頭に余分な空白を置くことを許す
*       先頭に'+','-'の符号をつけてもよい
*        ex) 123, +123, -123
*       '$','x','X'をつけると16進数とみなす
*        ex) $12AB, -$12AB, x12AB, X12AB
*
str      =       8
*
atoi:
         link    a6,#0                   *スタックフレーム生成
         movem.l d1-d3/a0,-(sp)          *{レジスタ待避
         movea.l str(a6),a0              *a0=文字列へのポインタ
         bra     atoi1

atoi0:   addq.w  #1,a0                   *文字列先頭の空白を
atoi1:   cmpi.b  #SPACE,(a0)             *  飛ばす
         beq     atoi0                   *
         cmpi.b  #TAB,(a0)               *
         beq     atoi0                   *

         moveq.l #1,d2                   *d2=符号(+)
         cmpi.b  #'+',(a0)               *'+'が指定されたか?
         beq     atoi2                   *
         cmpi.b  #'-',(a0)               *'-'が指定されたか?
         bne     atoi3                   *
         moveq.l #-1,d2                  *d2=符号(-)
atoi2:   addq.w  #1,a0                   *符号の分ポインタを進める

atoi3:   moveq.l #0,d0                   *結果を返すd0をクリア
         moveq.l #0,d1                   *作業用のd1をクリア
         moveq.l #-1,d3                  *仮にエラーフラグを立てる

         cmpi.b  #'$',(a0)               *16進の指定かどうか調べる
         beq     htoi                    *
         cmpi.b  #'X',(a0)               *
         beq     htoi                    *
         cmpi.b  #'x',(a0)               *
         beq     htoi                    *

         bra     atoi5
                                  *10進文字列のとき
atoi4:   addq.w  #1,a0
atoi5:   move.b  (a0),d1                 *1文字取り出す
         bsr     isdigit                 *数字か?
         bcs     atoiq                   *  そうでなければ終了
         mulu.w  #10,d0                  *10進1桁分左にシフト
         swap.w  d0                      *上位ワードが
         tst.w   d0                      *  0でなければ
         bne     atoie                   *  オーバーフローした
         swap.w  d0                      *
         add.w   d1,d0                   *下位に1桁追加
         moveq.l #0,d3                   *エラーフラグをクリア
         bra     atoi4                   *繰り返す

                                  *16進文字列のとき
htoi:    addq.w  #1,a0
         move.b  (a0),d1                 *1文字取り出す
         bsr     isxdigit                *16進数値か?
         bcs     atoiq                   *  そうでなければ終了
         asl.l   #4,d0                   *16進1桁分左にシフト
         swap.w  d0                      *上位ワードが
         tst.w   d0                      *  0でなければ
         bne     atoie                   *  オーバーフローした
         swap.w  d0                      *
         add.w   d1,d0                   *下位に1桁追加
         moveq.l #0,d3                   *エラーフラグをクリア
         bra     htoi

atoie:   moveq.l #-1,d3                  *オーバーフローが発生
atoiq:   muls.w  d2,d0                   *符号を掛けて最終結果を得る
         move.l  a0,str(a6)              *続く文字列へのポインタを返す
         tst.w   d3                      *エラーフラグをccrに反映する
         movem.l (sp)+,d1-d3/a0          *レジスタ復帰
         unlk    a6                      *スタックフレーム解放
         rts

*
*        10進文字→数値変換(d1.b)
*        (エラー時はC=1)
isdigit:
         subi.b  #'0',d1
         bcs     isdgtq
         cmpi.b  #9+1,d1
         CCF
isdgtq:  rts

*
*        16進文字→数値変換(d1.b)
*        (エラー時はC=1)
isxdigit:
         TOUPPER d1
         subi.b  #'0',d1
         bcs     isxdgq
         cmpi.b  #9+1,d1
         CCF
         bcc     isxdgq
         subq.b  #'A'-'0'-10,d1
         bcs     isxdgq
         cmpi.b  #15+1,d1
         CCF
isxdgq:  rts

         .end
```

リスト6　STAMP.S

```
         .include        doscall.mac
         .include        iocscall.mac
*
CFKEYMOD        equ     14      *CONCTRLモード番号
CSCREEN         equ     16      *
CCURON          equ     17      *
CCUROFF         equ     18      *
*
HIDEFKEY        equ     3       *ファンクションキー行非表示
DOS_GM3         equ     4       *画面モード512x512,256色
*
DISABLESKEY     equ     0       *ソフトウェアキーボード禁止
ENABLESKEY      equ     -1      *ソフトウェアキーボード許可
*
WINH            equ     272     *メニューウィンドウ幅
WINV            equ     104     *メニューウィンドウ高さ
*
USERPAGE        equ     1       *描画を行う画面
BIT_USERPAGE    equ     %0010   *
MENUPAGE        equ     0       *メニューを表示する画面
BIT_MENUPAGE    equ     %0001   *
*
                                *メニュー表示
SHOWMENU        equ     BIT_USERPAGE|BIT_MENUPAGE
                                *メニュー非表示
HIDEMENU        equ     BIT_USERPAGE
*
GAIJITOP        equ     $eb9f   *全角外字の先頭文字コード
FONT16          equ     $0008   *8x16,16x16
*
PATMAX          equ     8       *ペンパターンの最大数
*
*       グラフィック関係IOCSデータ受け渡し領域の構造
*
        .offset 0
*
X0:     .ds.w   1       *POINT  *FILL   *BOX
Y0:     .ds.w   1       *       *       *
RETCOL:                 *       *       *
X1:     .ds.w   1       *       *       *
POINTBUFSIZ:            *       *       *
Y1:     .ds.w   1               *       *
COL:    .ds.w   1               *       *
FILLBUFSIZ:                     *       *
LS:     .ds.w   1                       *
BOXBUFSIZ:                              *

*
*       フォント読み込み領域の構造
*
        .offset 0
*
XLEN:   .ds.w   1
YLEN:   .ds.w   1
FPAT:   .ds.w   16      *16x16
FNTBUFSIZ:
*
        .text
        .even
*
ent:
        lea.l   mysp(pc),sp             *spを初期化する

        bsr     init                    *画面などの初期化

        pea.l   break(pc)               *中断時の戻りアドレスを設定
        move.w  #_CTRLVC,-(sp)          *
        DOS     _INTVCS                 *
        move.w  #_ERRJVC,(sp)           *
        DOS     _INTVCS                 *
        addq.l  #6,sp                   *

        bsr     setupmenu               *メニューウィンドウの初期化
        bsr     main                    *メイン処理

break:  bsr     windup                  *後始末

        move.w  #-1,-(sp)               *キーバッファクリア
        DOS     _KFLUSH                 *
```


▶EXPERT-HDを母親が買った(私はACE-HD)。ポピュラスを妻が始めた。父親が森田将棋を始めた。我が家は楽しいX68000一家。アホラシー。　菊田　俊彦(31)茨城県

```
        addq.l  #2,sp                   *

        DOS     _EXIT                   *終了

*
*       メイン処理
*
main:
        IOCS    _MS_GETDT               *ボタンの状態を取得
        tst.b   d0                      *右ボタンが押されている?
        bne     rdown
        tst.w   d0                      *左ボタンが押されている?
        bpl     main
*
ldown:                          *左ボタンが押された
        IOCS    _MS_CURGT               *マウスカーソル位置を取得
        move.w  d0,d1                   *d1.w = y
        clr.w   d0
        swap.w  d0                      *d0.w = x
        tst.b   menuflag                *ウィンドウは表示中か?
        beq     pset

        move.w  winx(pc),d2             *d2.w = ウィンドウ表示位置x
        move.w  winy(pc),d3             *d3.w = ウィンドウ表示位置y

        cmp.w   d2,d0                   *ウィンドウ上かどうかチェック
        bcs     pset                    *
        cmp.w   d3,d1                   *
        bcs     pset                    *
        addi.w  #WINH,d2                *
        addi.w  #WINV,d3                *
        cmp.w   d2,d0                   *
        bcc     pset                    *
        cmp.w   d3,d1                   *
        bcc     pset                    *

                                *ウィンドウ内でクリックされた
        sub.w   winx(pc),d0             *d0.w = ローカルx座標
        sub.w   winy(pc),d1             *d1.w = ローカルy座標
        move.w  d0,pntbuf+X0            *x,yそれぞれを待避しておく
        move.w  d1,pntbuf+Y0            *

        subq.w  #8,d0
        bcs     drag                    *ウィンドウの左余白
        subq.w  #8,d1
        bcs     drag                    *ウィンドウの上余白
        cmp.w   #256,d0
        bcc     drag                    *ウィンドウの右余白
        cmp.w   #16,d1
        bcc     ldown1
                                *上段のメニュー内
        cmp.w   #224,d0
        bcs     ldown0
done:                           *終了ボックス内
        rts                             *メインループを抜ける

ldown0: subi.w  #32,d0
        bcs     drag                    *ペンメニューより左
        divu.w  #21,d0
        swap.w  d0
        cmpi.w  #16,d0
        bcc     drag                    *ペンパターンの隙間の余白
        swap.w  d0
*
selpen:                         *ペンメニュー内
        move.w  d0,d2                   *d2.w = ペン番号
        addi.w  #GAIJITOP,d0            *新パターンを設定
        move.w  d0,curpat               *

        moveq.l #MENUPAGE,d1            *メニュー用ページに
        IOCS    _APAGE                  *  切り換える

        lea.l   curpen(pc),a1           *以前の枠を消す
        move.w  #255,COL(a1)            *
        IOCS    _BOX                    *

        mulu.w  #24,d2                  *新たに枠を描く
        addi.w  #38,d2                  *
        move.w  d2,X0(a1)               *
        addi.w  #19,d2                  *
        move.w  d2,X1(a1)               *
        move.w  #1,COL(a1)              *
        IOCS    _BOX                    *

        bsr     lwait                   *左ボタンが離されるのを待つ

        bra     ldown2                  *描画用ページに戻す

ldown1: subi.w  #24,d1
        bcs     drag                    *ペンメニューと
                                        *  色メニューの隙間の余白
        cmpi.w  #64,d1
        bcc     drag                    *ウィンドウの下余白
*
selcol:                         *色メニュー内
        moveq.l #MENUPAGE,d1            *メニュー用ページに
        IOCS    _APAGE                  *  切り換える

        lea.l   pntbuf(pc),a1           *マウスカーソル位置から
        IOCS    _POINT                  *  色を拾う
        move.w  RETCOL(a1),d0           *
        move.w  d0,curcol               *カレントカラーにセット

        lea.l   coldat(pc),a1           *カレントカラーで
        move.w  d0,COL(a1)              *  メニュー左上の枠を
        IOCS    _FILL                   *  塗り潰す

ldown2: moveq.l #USERPAGE,d1            *描画用ページに戻す
        IOCS    _APAGE                  *

        bra     main                    *メインループへ
*
drag:                           *ウィンドウの外枠でクリックされた
        bsr     lwait                   *左ボタンが離されるのを待つ
        bra     menuon                  *ウィンドウを描き直す
*
pset:                           *ウィンドウ外でクリックされた
        lea.l   setdat(pc),a1           *マウスカーソルの位置に
        subq.w  #8,d0                   *  パターンを描く
        move.w  d0,X0(a1)               *
        subq.w  #8,d1                   *
        move.w  d1,Y0(a1)               *
        IOCS    _SYMBOL                 *

        bra     main                    *メインループへ
*
rdown:                          *右ボタンが押された
        move.l  #(WINH/2)<<16,ofst      *表示位置オフセット
        tst.b   menuflag                *メニューは表示中か?
        beq     menuon
*
menuoff:                        *メニューウィンドウを消す
        clr.b   menuflag                *フラグを寝かせる
        moveq.l #HIDEMENU,d1            *メニューページ非表示
        IOCS    _VPAGE                  *

        bsr     rwait                   *右ボタンが離されるのを待つ

        bra     main                    *メインループへ
*
menuon:                         *メニューウィンドウを出す
        st.b    menuflag                *フラグを立てる
        IOCS    _MS_CURGT               *マウスカーソル位置を取得
        move.w  d0,d1                   *d1.w = y
        swap.w  d0                      *d0.w = x

        sub.w   ofst+X0(pc),d0          *ウィンドウが
        bcc     mon0                    *  画面からはみ出さないよう
        clr.w   d0                      *  調整する
mon0:   sub.w   ofst+Y0(pc),d1          *
        bcc     mon1                    *
        clr.w   d1                      *
mon1:   cmp.w   #512-WINH,d0            *
        bcs     mon2                    *
        move.w  #512-WINH,d0            *
mon2:   cmp.w   #512-WINV,d1            *
        bcs     mon3                    *
        move.w  #512-WINV,d1            *

mon3:   move.w  d0,winx                 *表示位置を格納
        move.w  d1,winy                 *

        moveq.l #0,d2                   *ウィンドウを目的の位置へ
        sub.w   d0,d2                   *  移動する
        andi.w  #511,d2                 *
        moveq.l #0,d3                   *
        sub.w   d1,d3                   *
        andi.w  #511,d3                 *
        moveq.l #BIT_MENUPAGE,d1        *
        IOCS    _HOME                   *

        moveq.l #SHOWMENU,d1            *メニューページ表示
        IOCS    _VPAGE                  *

        bsr     rwait                   *右ボタンが離されるのを待つ

        bra     main                    *メインループへ

*
rwait:                          *右ボタンが離されるのを待つ
        IOCS    _MS_GETDT
        tst.b   d0
        bne     rwait
        rts
*
lwait:                          *左ボタンが離されるのを待つ
        IOCS    _MS_GETDT
        tst.w   d0
        bmi     lwait
        rts

*
*       初期化
*
init:
        link    a6,#0
                                *画面
        move.w  #DOS_GM3,-(sp)          *画面を512x512,256色に
        move.w  #CSCREEN,-(sp)          *  初期化
        DOS     _CONCTRL                *
        move.w  d0,scrnmsav             *現在の画面モードを待避

        move.w  #HIDEFKEY,-(sp)         *ファンクションキー行を
        move.w  #CFKEYMOD,-(sp)         *  非表示に設定
        DOS     _CONCTRL                *
        move.w  d0,fkeymsav             *現在のファンクションキー行
                                        *  モードを待避

        move.w  #CCUROFF,-(sp)          *カーソル非表示モード
        DOS     _CONCTRL                *

                                *外字
        bsr     savfont                 *待避
        bsr     deffont                 *定義

                                *マウス
        IOCS    _MS_INIT                *マウス初期化
        IOCS    _MS_CURON               *マウスカーソル表示
        moveq.l #DISABLESKEY,d1         *ソフトウェアキーボード
        IOCS    _SKEY_MOD               *  表示禁止

        unlk    a6
        rts

*
*       メニューの初期化
*       (あらかじめ全部描いておく)
*
setupmenu:
        moveq.l #MENUPAGE,d1            *メニュー用ページに
```

▶きぇ〜。68040が「自由にお持ち下さい」とは，なんと太っ腹なんでしょう……!?　マイコンショウに行けばよかった。ちぇっ。　　石渡　貴史(18)神奈川県

```
        IOCS    _APAGE                  * 切り換える
        moveq.l #HIDEMENU,d1            *メニュー用ページ非表示
        IOCS    _VPAGE                  *
        lea.l   fildat(pc),a1           *ウィンドウ枠を塗り潰す
        IOCS    _FILL                   *
        lea.l   boxes(pc),a1            *BOXを必要なだけ描く
boxlp:  tst.w   (a1)                    *
        bmi     boxed                   *
        IOCS    _BOX                    *
        lea.l   BOXBUFSIZ(a1),a1        *
        bra     boxlp                   *
boxed:
        lea.l   mendat(pc),a1           *ペンパターンメニューを
        IOCS    _SYMBOL                 * 描く
        bsr     makecoltbl              *カラーテーブル
        moveq.l #USERPAGE,d1            *描画用ページに切り換える
        IOCS    _APAGE                  *
        clr.b   menuflag                *フラグを寝かせる
        rts
*
*       256色の色テーブルを描く
*
makecoltbl:
        link    a6,#-FILLBUFSIZ
        lea.l   -FILLBUFSIZ(a6),a1      *a1=FILL用パラメータ領域
        moveq.l #0,d1                   *d1=色
        move.w  #32,Y0(a1)              *(8,32)-(8+7,32+7)から
        move.w  #32+7,Y1(a1)            *
        moveq.l #8-1,d6                 *縦に8個
clp0:   move.w  #8,X0(a1)
        move.w  #8+7,X1(a1)
        moveq.l #32-1,d7                *横に32個
clp1:   move.w  d1,COL(a1)              *四角を描く
        IOCS    _FILL
        addq.w  #8,X0(a1)               *右に8ドット移動
        addq.w  #8,X1(a1)               *
        addq.w  #1,d1                   *次の色
        dbra    d7,clp1                 *横1列分繰り返す
        addq.w  #8,Y0(a1)               *下に8ドット移動
        addq.w  #8,Y1(a1)               *
        dbra    d6,clp0                 *繰り返す
        unlk    a6
        rts
*
*       後始末
*
windup:
        link    a6,#0
        move.w  scrnmsav,-(sp)          *画面モードを戻す
        move.w  #CSCREEN,-(sp)          *
        DOS     _CONCTRL                *
        move.w  fkeymsav,-(sp)          *ファンクションキー行の
        move.w  #CFKEYMOD,-(sp)         * モードを戻す
        DOS     _CONCTRL                *
        move.w  #CCURON,-(sp)           *カーソル表示モード
        DOS     _CONCTRL                *
        bsr     rstfont                 *外字フォント復帰
        IOCS    _MS_INIT                *マウス初期化
        moveq.l #ENABLESKEY,d1          *ソフトウェアキーボード
        IOCS    _SKEY_MOD               * 表示許可
        unlk    a6
        rts
*
*       外字の先頭8文字のフォントパターンを待避する
*
savfont:
        lea.l   fontbuf(pc),a1
        move.l  #FONT16<<16|GAIJITOP,d1
        moveq.l #PATMAX-1,d2
savlp:  IOCS    _FNTGET
        addq.w  #1,d1
        lea.l   FNTBUFSIZ(a1),a1
        dbra    d2,savlp
        rts
*
*       外字の先頭8文字にフォントパターンを設定する
*
deffont:
        lea.l   fontdat+FPAT(pc),a1
defnt0: move.l  #FONT16<<16|GAIJITOP,d1
        moveq.l #PATMAX-1,d2
deflp:  IOCS    _DEFCHR
        addq.w  #1,d1
        lea.l   FNTBUFSIZ(a1),a1
        dbra    d2,deflp
        rts
*
*       savfontで待避したフォントパターンを復帰する
*
rstfont:
        lea.l   fontbuf+FPAT(pc),a1
        bra     defnt0
*
*       データ&ワーク
*
        .data
        .even
*
fontdat:
        .dc.w   16,16                   *eb9f
        .dc.w   %0000000000000000       *0
        .dc.w   %0000000000000000
        .dc.w   %0000000000000000
        .dc.w   %0011111110111111
        .dc.w   %0010000001010001
        .dc.w   %0001000001010010
        .dc.w   %0001000000101100
        .dc.w   %0000100000101000
        .dc.w   %0000100000010000
        .dc.w   %0000010000010000
        .dc.w   %0000010000001000
        .dc.w   %0000101000001000
        .dc.w   %0001101000000100
        .dc.w   %0010010100000100
        .dc.w   %0100010100000010
        .dc.w   %0111111011111110
        .dc.w   16,16                   *eba0
        .dc.w   $0000,$0000,$0000,$0000,$0000,$0000,$0000,$0000
        .dc.w   $0080,$0000,$0000,$0000,$0000,$0000,$0000,$0000
        .dc.w   16,16                   *eba1
        .dc.w   $0000,$0000,$0000,$0000,$0000,$0000,$0000,$0180
        .dc.w   $0180,$0000,$0000,$0000,$0000,$0000,$0000,$0000
        .dc.w   16,16                   *eba2
        .dc.w   $0000,$0000,$0000,$0000,$0000,$0000,$0000,$01c0
        .dc.w   $01c0,$01c0,$0000,$0000,$0000,$0000,$0000,$0000
        .dc.w   16,16                   *eba3
        .dc.w   $0000,$0000,$0000,$0000,$0000,$0410,$0220,$0140
        .dc.w   $0080,$0140,$0220,$0410,$0000,$0000,$0000,$0000
        .dc.w   16,16                   *eba4
        .dc.w   $0000,$0000,$0000,$0000,$0000,$0c00,$0600,$0300
        .dc.w   $0180,$00c0,$0060,$0030,$0000,$0000,$0000,$0000
        .dc.w   16,16                   *eba5
        .dc.w   $0000,$03e0,$07f0,$0ff8,$1ffc,$3ffe,$7fff,$7fff
        .dc.w   $7fff,$7fff,$7fff,$3ffe,$1ffc,$0ff8,$07f0,$03e0
        .dc.w   16,16                   *eba6
        .dc.w   $0000,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff
        .dc.w   $7fff,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff
fildat: .dc.w   0,0                     *ウィンドウ枠塗り潰し用
        .dc.w   WINH-1,WINV-1
        .dc.w   255
boxes:                                  *ウィンドウ内枠描画用
box1:   .dc.w   2,2,WINH-2-1,WINV-2-1,1,$ffff
box2:   .dc.w   6,6,33,25,1,$ffff
curpen:
box3:   .dc.w   38,6,57,25,1,$ffff      *カレントペンを示す枠
box4:   .dc.w   230,6,265,25,1,$ffff
box5:   .dc.w   6,30,265,97,1,$ffff
        .dc.w   -1
mendat: .dc.w   40,8                    *メニュー表示用
        .dc.l   patstr
        .dc.b   1,1
        .dc.w   1
        .dc.b   1,0
coldat: .dc.w   8,8,31,23,255           *カレントカラー表示用
setdat: .dc.w   0,0                     *点描画用
        .dc.l   curpat
        .dc.b   1,1
curcol: .dc.w   255
        .dc.b   1,0
curpat: .dc.b   $eb,$9f,0               *カレントペンパターン
patstr: .dc.b   $eb,$9f,$20,$eb,$a0,$20 *メニュー文字列
        .dc.b   $eb,$a1,$20,$eb,$a2,$20
        .dc.b   $eb,$a3,$20,$eb,$a4,$20
        .dc.b   $eb,$a5,$20,$eb,$a6,$20
        .dc.b   '終了',0
*
        .bss
        .even
*
fontbuf:        .ds.b   FNTBUFSIZ*8     *フォント待避領域
ofst:
pntbuf:         .ds.b   POINTBUFSIZ     *IOCS POINT用
winx:           .ds.w   1               *メニューウィンドウ表示位置
winy:           .ds.w   1               *
scrnmsav:       .ds.w   1               *画面モード待避用
fkeymsav:       .ds.w   1               *ファンクションキー行モード
                                        * 待避用
menuflag:       .ds.b   1               *メニュー表示/非表示フラグ
*
        .stack
        .even
*
mystack:
        .ds.l   1024                    *スタック領域
mysp:
        .end    ent
```

PASCALのデータ型を見る

PASCALはさまざまなデータを多彩な方法で取り扱うことができます。それは整数や実数などの数値，文字列といったものから集合やポインタにまで及びます。それではPASCAL言語におけるデータの扱い方をまとめて見てみましょう。

Fujii Yoshimi/Fujiki Takeshi
藤井義巳/藤木健士

連載も3回目になりますが，読者の皆さんはもうPASCALのプログラムをいくつか書いてみられたことと思います。Cを知っている人なら「なぁんだ，簡単じゃねーか」と思われたでしょう。Cを知らない人でもそれほど難しくないですよね。ただ，PASCALは型の厳しい言語なので，型についてよく知っておかないとしょっちゅうコンパイラから文句をいわれます。たとえばCなら整数変数に実数値を代入しても，勝手に変換してくれていたのに，PASCALではエラーになるといった具合です。そこで今月はその型について，少し詳しく説明することにします。

データ型

WirthはPASCALをプログラミングの教育に使いたいと考えました。彼は著書『アルゴリズム＋データ構造＝プログラム』で，プログラムを作る際のデータ構造の大切さを教えています。その彼が設計したPASCALが豊富なデータ型を備えていたのは当然のことで，さまざまなデータ構造を直接に記述することができます。PASCALのデータ型はおおまかに単純データ型と構造データ型，およびポインタ型に分類することができます。単純データ型はさらに，整数型，実数型，列挙型，論理型，文字型，部分範囲型に分かれます(図1)。また，実数型以外の単純データ型は順序型とも呼ばれて，共通の特徴を持っています。構造データ型には配列型，ファイル型，集合型，レコード型があります。PASCALの構造データ型はPackedという形容詞をつけると，多少の速度を犠牲にしても主記憶を食わないように，詰め込んだデータ型になります。このあたりは処理系によって対応がまちまちですが，後述の文字列型に関しては必ずPackedと書かなければならないことになっています。

PASCALがデータ構造の表現力に優れているのは，レコード型とポインタ型のおかげです。リスト構造，ツリー構造，キューなどの基本的なデータ構造をレコード型とポインタなしで表現することを想像してみてください。なにを隠そうFORTRANの世界では，21世紀を迎えようとする今日になっても，「データ構造はすべて配列で作る」なんて野蛮なことが行われているのです。信じられませんね。

前置きはこれくらいにして，それぞれの型について説明していきましょう。

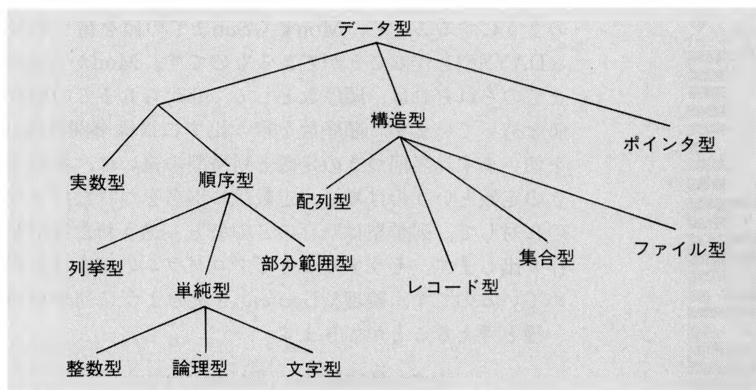
整数型と実数型

整数型と実数型はどの言語でもお馴染みの型ですね。PASCALの整数型の最大値は定義済み定数Max Intで知ることができます。例によって8086系CPUの処理系ではMax Intは32767であることが多く，PurePASCALは $2^{31}-1$ (計算して！)です。実数についてはなにもいうことはありません。式の中で実数と整数は混在して使うことが許されています。というより，整数は実数が必要とされる文脈では自動的に実数に変換されます。整数変数に実数値を代入することはできません。実数から整数に変換する方法はあとで説明します。整数型はInteger，実数型はRealという名前です。PurePASCALではReal型は32ビットの単精度のみ用意されています(PurePASCALは実数演算にFLOAT??.Xを利用しています。FLOAT2+.XとFLOAT3+.Xは，それぞれFLOAT2.X，FLOAT3.Xよりも速いのですが，単精度浮動小数点演算にバグを持っており，PurePASCALでの実数演算でおかしな結果が出ることがあります)。

論理型

C言語やBASICでは論理式の値が整数になっていましたが，PASCALには独立した論理型が存在します。論理型は型名Booleanで定義され，TrueかFalseかどちらかの値を取ります。たとえばa=1，b=1のとき，式a=bはTrueで，a=−1，b=1のとき，式a>bはFalseです。

図1 PASCALのデータ型



文字型

文字型の変数には,

```
    var c:char;
  begin
          ⋮
    c:='A';
          ⋮
  end
```

といったようにキャラクタコードが格納されます。Cの文字型は8ビットの整数として使われていましたが,PASCALでは独立した型で,整数との混用はできません。

文字列は後ろでも説明するつもりですが,文字型のPACKED配列として作ります。文字型の定数は'A'のように表現します。漢字などの2バイトコードは文字型には使えません。PurePASCALでは2バイトコードは,文字列の中でだけ使用可能です。

列挙型

列挙型はC言語でもお馴染みですね。知らない人のために少し説明します。プログラム中に定数値をマジックナンバーとして埋め込むと,わかりにくくなりがちです。

```
  if data=-1 then
```

と書くよりも,

```
  const ILLEGAL=-1;
```

と定数を定義しておいて,

```
  if data=ILLEGAL then
```

と書いたほうがよいことはわかりますね。こうすると,あとで仕様変更してILLEGALの値が変わったときでも,最初の定数定義を変更するだけですみます。

この名前つきの定数と似た概念として,列挙型というものがあります。これは,

```
  type DAYS=(Mon, Tue, Wed, Thu, Fri, Sat,
  Sun);
  var date:DAYS;
```

のようにすることで,MonからSunまでの値を持つ新たなDAYS型を作ることができるものです。MonからSunまでのそれぞれは,順序数として,0から6までの整数値を持っています。順序数を取り出すには標準関数Ordを使います。名前つきの定数と列挙型の違いは,名前つきの定数というのは単に,定数値に別名をつけただけなのに対して,列挙型は既存のどの型とも違う新たな型を作り出します。そうすることでプログラムの安全性を高めているのです。論理型Booleanも次のように列挙型の一種と考えることができます。

```
        type Boolean=(False, True);
```

よってOrd(False)=0,Ord(True)=1です。

部分範囲型

部分範囲型は,事前に定義された順序型のある範囲の値だけをとる型です。たとえば,

```
  type Subrange=1..10;
       Weekday=Mon..Fri;
  var  i:Subrange;
       d:Weekday;
```

という具合に部分範囲型Subrange,Weekdayを定義すると,Subrange型の変数iは1から10まで,Weekday型の変数dはMonからFriまでの値だけを代入することができるわけです。それ以外の値を代入したら実行時エラーになるでしょう。

このようなチェック機構もPASCAL系の言語の特徴です。バグのために変数に予期せぬ値が代入されるなんてよくある話です。こんなとき,Cだったらデバッグ用のprintfをたくさん入れて再コンパイルということを皆さんしているのではないでしょうか。ソースコードデバッガがあれば少しはましでしょうが,それにしてもバグの箇所を特定するのにかなり苦労するでしょう。PASCALだったら多くの場合,ランタイムエラーで一発で見つかるわけです。C言語もこういった機構を取り入れて,

```
  int i:1..10;
```

なんて記述ができればいいと思いませんか。チェックはコンパイルスイッチでいつでもoffできるわけですからね。それにいま,shortとかlongとか,ユーザーが決定しているのですが,このように書けたらコンパイラが勝手にintのサイズを決めてくれるわけです。いまCコンパイラを作っている人がいましたら,ぜひ考えてください。規格のあと追いだけじゃ面白くないでしょ。なんて書きましたが,PurePASCALもそのあたりはさぼっていて,部分範囲型も必ず4バイトを占めます。

配列型

配列型の定義は,

　type 配列型名=array [添字型] of要素型;

というかたちで行われます。要素型は任意の型,添字型は任意の順序型が指定できます。いくつかの例を挙げると次のようになります。

```
  type arrayA=array [-1..10] of Real;
       arrayB=array[Boolean]of array[Char]
  of Integer;
       arrayC=array [1..10, 1..10] of Real;
       arrayD=array [DAYS] of Integer;
```

arrayAは添字の下限が-1,上限が10,要素が12の実数配列です。arrayBは2×256要素,arrayCは10×10要素の2次元配列です。

arrayCの表記は,

PurePASCAL

```
type arrayC=array [1..10] of array [1..10]
of Real;
```

のかたちの省略形です。つまり，多次元配列は「配列の配列」と解釈されます。

```
var A:arrayA;
    B1,B2:arrayB;
    C1,C2:arrayC;
    i:Integer;
    date:DAYS;
    E:array [−1..10] of Real;
```

のとき，

```
A [i] :=1.23;
B1 [False] :=B2 [True] ;
C1:=C2;
for date:=Mon to Sun do
    arrayD [date] :=0;
```

といったような操作が可能です。Cとは違って配列の代入が可能で，配列のすべての要素がコピーされます。また，iの値が−1..10のあいだにないときは実行時エラーとなります。

ここでひとつ注意しなければならない点があります。変数Aと変数Eは一見すると同じ型のように見えるのですが，

```
E:=A;
```

のような操作は許されません。実はAとEは別々の型なのです。もっと極端な例を示すと，

```
var va:array [1..10] of Integer;
    vb:array [1..10] of Integer;
```

このとき，vaとvbは別々の型になってしまうのです。もしvaとvbを同じ型にしたいなら，

```
var va, vb:array [1..10] of Integer;
```

または，

```
type array1:array [1..10] of Integer:
var  va:array1;
     vb:array1;
```

と書いてください。このように，PASCALの型が同一かどうかの判断は，その構造で判断するのではなく，型の名前かまたは変数が宣言された場所で決まるのです。「面倒でも型には名前をつけろ」ということなのでしょう。これは配列に限ったことではなく，ほかの構造型でも同じです。気をつけましょう。

文字列型

PASCALには文字列のための特別な型は存在しません。Cと同じように，文字列は文字型の配列として表されます。もう少し厳密な定義を示します。

```
type StringN=Packed array [1..N] of Char;
var str:StringN;
```

ただし，1<Nという約束です。'Packed'を忘れてはいけません。このように定められた文字列型に関しては，同じ文字列型同士の代入，関係演算(inを除く)が許されています。この「同じ文字列型」というのは，簡単にいえばNが同じということです。長さが違う文字列型同士の演算，代入はできません。あまり融通がききませんね。文字列定数は'(single quote)で囲んで表現します。

```
Const N=25:
type  StringN=Packed array [1..N] of Char;
var   str1, str2:StringN;

begin
      str1:='This is character string.';
      str2:='This is string too.       '
end;
```

このように，文字列定数の長さが短いときは，帳尻合わせにスペースを入れてください。

集合型

集合型はPASCAL独特のデータ型です。順序型の値の集合をビット列で表現し，集合演算を行うことができます。ただし，PurePASCALではかなりの制限つきで，集合の要素は順序数が0〜127のものしか許されません。つまり，集合変数が128ビット=16バイトで表現されるわけです。私自身はあまり利用しないのですが，集合演算が直接行えるのは便利なのかもしれません。集合型の定数は［　］の中に，要素を,(カンマ)で区切って並べます。また，要素が連続している場合は途中の要素を全部書く代わりに［3，10..20，40］といった記述もできます。

例：

```
type SetOfDays=Set of DAYS;
var weekday, allday:SetOfDays;
begin
    allday:= [Mon..Sun] ;
    weekday:=allday− [Sat,Sun]
end;
```

レコード型

C言語の構造体に当たるもので，いくつかの変数をまとめてひとつの変数として扱うデータ型です。このレコード型と，次に説明するポインタ型を組み合わせて，PASCALは実にさまざまなデータ構造を表現できます。

レコード型の定義は，

```
type レコード型名=record固定部　可変部　end;
```

のようになされます。可変部とは，C言語でいう共用体にあたるものです。ありふれた例ですが，複素数型の作り方を例にレコード型の説明をしましょう。

```
type Complex＝record
        Re, Im;Real
    end;
var a,b,c:Complex;
begin
    c.Re:＝a.Re＊b.Re－a.Im＊b.Im;
    c.Im:＝a.Re＊b.Im＋a.Im＊b.Re
end;
```

レコード型Complexは2つのフィールドRe，Imを持っています。ReとImはそれぞれReal型です。Complex型の変数aのフィールドReをアクセスするにはa.Reのように，変数名のあとに.とフィールド名Reを書きます。また，レコード型の変数同士の代入もできます。レコード型の構文規則はかなり複雑なので，構文図をつけておきますから参考にしてください（図2）。

ポインタ型

ポインタ型の変数は対象型と呼ばれる型の変数のアドレスを保持します。C言語のポインタと違って，アドレスの値を整数値に変換したり，あるいはポインタ同士，ポインタと整数のあいだで演算したりすることはできません。また，配列とも関係ありません。ポインタに対して許される演算は，同じポインタ同士の一致と不一致，それからポインタがなにも指していないことを意味する定数Nilとの比較だけです。ポインタ型の定義は，

```
type DataPtr＝＾Data;
    Data＝Record
        item:Real;
        next:DataPtr
    end;
```

のように，対象型の前に＾をつけます。また，対象型にアクセスするときは，ポインタ変数の後ろに＾をつけて行います。標準手続きNew(p)は対象型の変数の領域を主記憶上に確保し，そのアドレスをポインタ変数pに格納します。不要になった領域は標準手続きDispose(p)で解放します。図3に双方向リスト，二進木を図示します。これらのデータ構造は次のデータ型Tree，Listで表現できます（よく見たらTreeもListも型の構造は同じですね）。

```
type TreePtr＝＾Tree
    Tree＝Record
        data:Item;
        left, right:TreePtr
    end;
    ListPtr＝＾List
    List＝Record
        data:Item;
        prev,next：ListPtr
    end;
```

ファイル型

ファイル型というのは文字どおりファイルを使うために用意されたデータ型です。ただ，PASCALのファイルの概念は現在のUNIXやMS-DOSのそれとは大きく隔たりがあって，そのままではあまり実用的ではないと思います。詳しくは参考文献を読んでいただくことにして，ここでは概念を示すにとどめます。

ファイルは次のように宣言されます。

```
var DataFile:File of Data;
```

PASCALのファイルはシーケンシャルファイルで，ファイル処理はいくつかの標準手続きによって行われます。標準手続きを簡単に説明すると，次のようになります。

Reset(f)	ファイルfを読み込みのために初期化する。
Rewrite(f)	ファイルfから要素をひとつ取り，その値をf＾に入れる。
Get(f)	ファイルfから要素をひとつ取り，その値を変数f＾に入れる。

図2　レコード型の構文

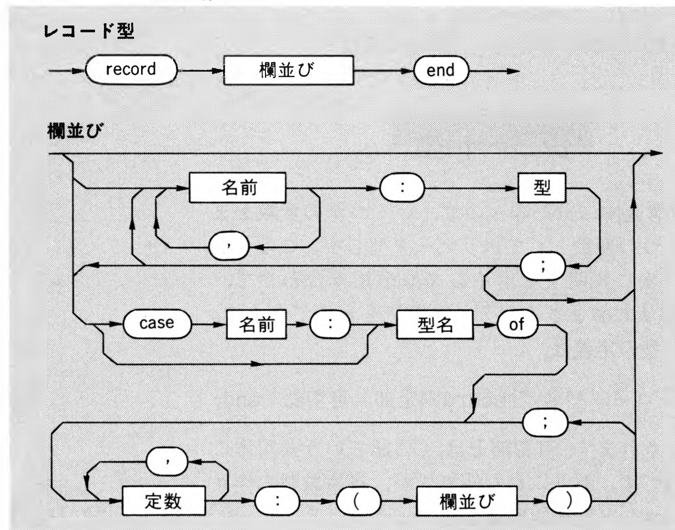
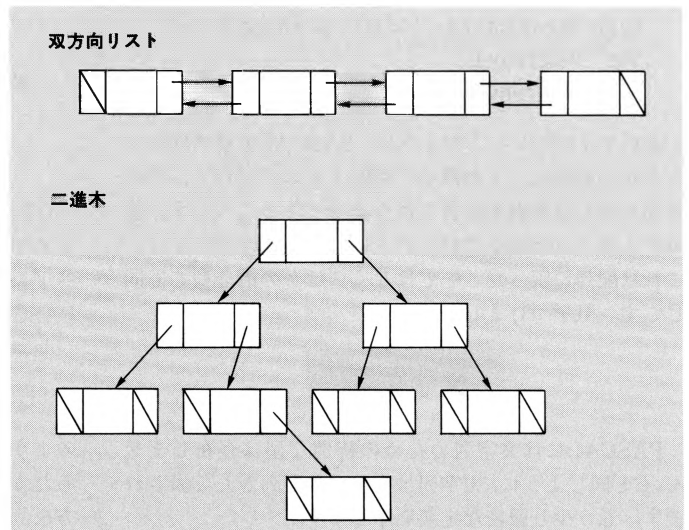


図3　データ構造の例



Put(f)　　　バッファ変数f^の値をファイルfに書き込む。

Read(f,x)　　ファイルfから要素をひとつ取り，その値を変数xに入れる。

Write(f,x)　　xの値をファイルfに書き込む。

fがファイル変数のとき，f^（ポインタみたい）はバッファ変数と呼ばれ，ファイルを読み書きする場合，要素ひとつ分だけのバッファとなります。また，手続きRead，Writeは例外的に不定個の引数を取り，複数の要素を一度に読み書きできます。

```
write(f1,x1,x2,x3);
read( f 2,y1,y2,y3);
```

は，

```
write(f1,x1);write(f1,x2);write(f1,x3);
read(f2,y1);read(f2,y2);read(f2,y3);
```

と同じことです。

プログラムの最初に，

```
program main(input, output);
```

と書きますが，このinputとoutputもファイル型の変数です。readとwriteの最初の引数を省略すると，readはinput，writeはoutputが指定されたと解釈されます。また，inputとoutputはテキストファイルと呼ばれる特殊なファイルで，readとwriteはこのテキストファイルに関しては特別な振る舞いをします。詳しくは次の機会に譲ります。

標準関数

PASCALにはいくつかの標準関数が用意されています。PASCALの標準関数はほかの言語と比較すると非常に少ないですが，教育用としては十分でしょう（表1）。それよりも問題となるのは，PASCALの標準関数のいくつかは，PASCAL自身で作ることができないという点です。たとえばPredとSuccは任意の順序型の値を引数として取れることになっていますが，PASCALの言語仕様では引数の型を複数指定することは許されていません。このような汚い点があることをWirthは素直に認めていて，Modula-2では改善したようです。

Chrは一種の変換関数なのですが，C言語で言うところのキャスト演算子と解釈することもできます。事実，MacintoshのTHINK PASCALでは，Cだったら，

　（型名）x

と書くところを，

　型名（x）

のかたちで，xの型を“型名”で示される型に変換することができるようです。整数型から文字型への変換が，ChrじゃなくてCharだったら完璧だったのにね。

実数を引数に取る関数は，整数も引数にできます。なぜなら整数は実数に自動的に変換されるからです。逆に実数→整数の変換は明示的に行う必要があり，そのために2種類の関数（RoundとTrunc）が用意されています（表2）。

表1　PASCALの標準関数

関数	引数	戻り値	説明
Abs(x)	実数型または整数型	実数型または整数型(xと同じ)	xの絶対値
ArcTan(x)	実数型	実数型	xの正接
Chr(i)	整数型	文字型	整数型→文字型変換
Cos(x)	実数型	実数型	xの余弦
Eof(f)	ファイル型	論理型	End of File
Eoln(f)	ファイル型	論理型	End of Line
Exp(x)	実数型	実数型	xの指数関数
Ln(x)	実数型	実数型	xの自然対数
Odd(i)	整数型	論理型	iが奇数なら真
Ord(o)	順序型	整数型	順序型→整数型
Pred(o)	順序型	oと同じ順序型	ひとつ前の要素を得る
Round(x)	実数型	整数型	xを四捨五入
Sin(x)	実数型	実数型	xの正弦
Sqr(x)	実数型	実数型	xの自乗
Sqrt(x)	実数型	実数型	xの2乗根
Succ(o)	順序型	oと同じ順序型	次の要素を得る
Trunc(x)	実数型	整数型	xを切り捨て

表2　RoundとTrunc

Round(x)	0≦xのとき，x＋0.5以下の最大の整数
	x＜0のとき，x－0.5以上の最小の整数
Trunc(x)	0≦xのとき，x以下の最大の整数
	x＜0のとき，x以上の最小の整数

区別が面倒くさいかもしれませんが，0≦xのときは，Roundが四捨五入（丸め），Truncが切り捨てと覚えておけば十分でしょう。

Ord(o)は順序型の順序数を調べる関数です。oが整数のときは当然o＝Ord(o)が成り立ちます。ASCIIコードを採用している処理系ならば（PurePASCALはもちろん）Ord（'A'）＝65ですね。

また，列挙型の場合，

```
type Colors=(Red, Blue, Green);
```

となっていたら，Ord(Red)＝0，Ord(Blue)＝1，Ord(Green)＝2となるでしょう。PASCALにはこのOrdの逆関数がないのも困りものです。このときPred(Blue)＝Red，Succ(Blue)＝Greenということになります。Pred（Red）やSucc（Blue）はエラーです。論理型は，

```
type Boolean= (False, True) ;
```

と考えられるので，Ord（False）＝0，Ord（True）＝1となります。

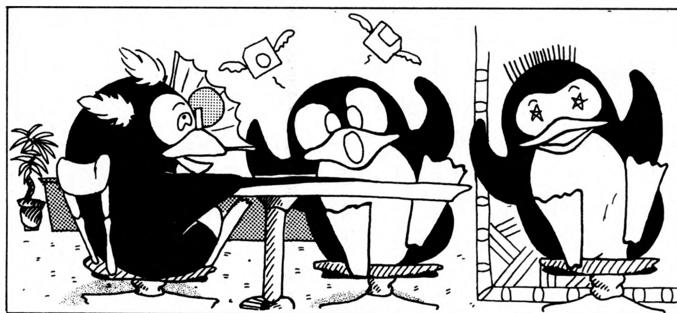
＊　　＊　　＊

今月はPASCALの持つ豊富なデータ型について説明しました。本当は演算子の説明までやりたかったのですが，型の説明だけでかなり長くなってしまいました。演算子については次の機会に説明することにしましょう。

参考文献

[1] Niklaus Wirth：“Algorithms＋Data Structures＝Programs”, Prentice-Hall, 1976（邦訳）片山卓也：「アルゴリズム＋データ構造＝プログラム」，日本コンピュータ協会，1979

[2] Per Brinch Hansen:“Programming a Personal Computer”, Prentice-Hall, 1983（邦訳）玄光男：「パソコンシステムプログラム設計　I．コンパイラ設計編」，電気書院，1988



マシン語カクテル　in Z80's Bar

第14回──楽な逆ポーランド？──

シナリオ＆イラスト：山田純二
特別監修：浦川博之　金子俊一

前回では演算ルーチンと変換部分のみの発表しかできなかった電卓でしたが，今回はいよいよその完成版が登場。さあ，はたしてこれでツケは払うことができるのか。緊張感あふれる展開。なーんちゃって，そんなことあるわけないでしょ。

カランコロ〜ン！

マスター（以下M）：いらっしゃいませ。

純二（以下**純**）：どうぼ，ごんばんばあ〜（どうも，こんばんは）。

ようこ（以下Yo）：あら，純二君おひさしぶり。どうしたの？　やばせばびのものまね？

長老（以下**老**）：それにしてもへたくそじゃのう。もっと修業を積まんとお笑い芸人にはなれんぞ。

純：ぶたりども，びどいいいがたでずね。ばぐだって，ずぎでやってるんじゃないんでず（２人とも，ひどい言い方ですね。僕だって，好きでやってるんじゃないんです）。

善司（以下**善**）：趣味でやってんでしょ。

老：同じじゃ，ばかもの。

純：がぜをびいでしまったんでずよ（かぜをひいてしまったんですよ）。

M：どうせ徹夜でポピュラスでもやっていたんでしょう。

純：おおばたび！　どいいだいでずが，じつばしんがんごんぱで，よどおじざわぎまぐって，あざおぎでみだらごうなっでいだんでず。（大当たり！　と言いたいですが，実は新歓コンパで夜通し騒ぎまくって，朝起きたらこうなっていたんです）。

老：理由はわかったが，そのしゃべり方はなんとかならんか。

純：ぜりぶがにばいばじになっでげんごうがずずんでいい，どらいだーばおもっでいぬようでずが（セリフが２倍増しになって原稿が進んでいい，とライターは思っているようですが）。

M：そんなことよりツケの残りを早く払ってくださいよ。

純：えっ，なにいってんです。先月はプログラムをちゃんと渡したじゃないですか（いきなり元の声に戻った）。

Yo：残念。前回の騒ぎ，全部純二君もちになっているわよ。

純：そんな，バナナん，ばななん，ば，な，な。

善：空にキラキラお星様……。

Yo：あなたは寝ていなさい。

善：ぐうぐう……。

老：増える増える飲み屋のツケ。大きくなれよ。

純：大きくなってたまりますか。長老が女の子の分はもって，あとはワリカンという話はどうなっちゃったんですか？

老：さーて，なんのことかな。わしゃ知らんぞ。

純：まったく，これだから年寄りは嫌なんだよな。

老：ほっほっほ。いくら反論してもツケは消えん，かんねんせい。

純：とほほ……。



電卓プロジェクト始動

老：ま，ツケの話はともかく，前回の続きをやってもらおうかの。

M：電卓のプログラムですよね。

純：そうです。ちょっとリストが大きかったので，今月まで残っちゃったんです。

老：数式をちゃんと記述できる電卓を作っておったんじゃったな。その式の計算方法はどうやっておるのじゃ。

純：それはですね。式をいったん逆ポーランド記法に変換してから計算して答えを求めています。

Yo：ふーん。ポーランド人が逆立ちでもするの？

純：いや，その場合ドンラーポ記法といったほうが……。

老：……逆ポーランド記法というのはじゃな。演算子には優先順位があるということを考えてできた計算法のことじゃよ。

Yo：優先順位って掛け算と割り算は足し算，引き算よりも先にやるっていうアレのこと？

純：そう。順次処理の好きなコンピュータには逆ポーランド記法が便利なんですよ。

Yo：へえ。なんで？

老：たとえば，10＋５×３という式は，まず５×３を計算してから10を足すじゃろう。こんなふうに式の中を先へ進んだり，元に戻ったりするとプログラムがややこしくなってしまうのじゃ。

純：それが逆ポーランド表記ならさっきの式を例にとってみると，

　10　５　３　＊　＋

と表せるんですよね。

Yo：ただ順番を並べ替えたってだけじゃないの。

老：これにはちゃんとした意味があるのじゃよ。純二君や，これを実際に計算することができるかの。

純：もちろん。計算にはスタックを使うんです。左から式の内容を調べて定数はスタックに積んでいき，演算子が見つかったらスタックから定数２つを取り出し計算します。その答えをまたスタックに積み直してこれを終わりまで繰り返せば出来上がり。

老：正解じゃ。このように逆ポーランド記法を使えば，式の中をいったりきたりせずに左から順番に処理することができるのじゃよ。わかったかな，ようこちゃん。

Yo：なるほど，便利にできてるのね。



どうすりゃいいんだ

M：では，逆ポーランド記法への変換のアルゴリズムはどうなっているんですか。

純：まず，式変換ワーク，演算ワークと，このプログラムの心臓部といえる変換テーブル（リストの682〜694行参照）を用意し

ます。演算子をどういう順番で書くかはこの変換テーブルが決めてくれるんです。

で，左から順番に式の項を読んでいき，定数は式変換ワークに出力。演算子だったらいま取り出した演算子を横の値，演算ワークのいちばん新しい演算子を縦の値として変換テーブルの内容を取り出します。この内容に従って次のような処理をしていきます。

00　演算ワークのいちばん新しいものを取り出して式変換ワークに出力。取り出した演算子はそのままで同じ処理を繰り返す。

01　取り出した演算子を演算ワークに出力。

02　演算ワークのいちばん新しいものを捨てて，新たに“＊”を演算ワークに出力する。

03　演算ワークのいちばん新しいものを，ただ捨てる。

04　演算ワークのいちばん新しいものを捨てて，新たに“＊”を２つ演算ワークに出力する。

05　終了

99　エラー

と，どんどん処理を繰り返していくと，式変換ワークに逆ポーランド記法の式が出来上がるわけです。

Yo：そんなふうにいわれても……。

純：わかりました。それじゃあ，

10＋20＋30＊40－50

という式を変換していく様子を見ていきましょう。まず，10は式変換ワークに出力。次は演算子の“＋”がくるので変換テーブルの内容を取り出しにいきます。テーブルの縦の値は“＋”で，横の値は演算ワークのいちばん新しいもの（この場合はなにも入っていないからワークのエンドコード）。処理内容は01だから，“＋”をそのまま演算ワークに出力します。

次の項は20。定数項だから式変換ワークに出力。２番目の演算子“＋”のときも１番目と同様にしてテーブルの中身を見る。縦も横も“＋”の場合，処理内容は00なので演算ワークのいちばん上の演算子を式変換ワークに出力する。この場合は式から取り出してきた演算子はそのままで，もう一度変換テーブルを見る。今度は縦が“＋”，横がワークのエンドコードで処理内容は01となるので演算ワークに“＋”を出力する。ここまでで式変換ワークと演算ワークにどのような出力がされているかわかりますか，ようこさん。

Yo：式変換ワークには10と20に最初の“＋”があって，演算ワークには２番目の“＋”があると思うけど。

純：正解ですよ，ようこちゃん。これでだいたいの流れはつかめたでしょう。

老：それで最後には，

10　20　＋　30　40　＊　＋　50　－

という式に変換されるわけじゃな。

Yo：なるほどね。

カッコがつくぞ～

M：純二君，いままでの話を聞いているとわざわざテーブルとか用意しなくてもプログラムで何とかなりそうな気がするけど。

純：確かにやっていることは優先順位に従って２つの処理を選択しているだけですからね。しかし，この変換テーブルのおかげで括弧を使った式の展開のプログラムがすっきり組めるんですよ。

老：まあ，テーブルを使わずにやってやれないことはないがリストは汚くなってしまうじゃろうな。

Yo：リストは読みやすいにこしたことはないってことね。

純：じゃ，話も一段落したところで，次はその括弧の話。括弧の使い方には３通りのパターンがあります。

１番目は，10＋(20＋30）のように，括弧の中の式をただ優先させるもの。

２番目は，10(20＋30)または(20＋30)10のように，前か後ろのどちらかの括弧に乗算が省略されている場合。

３番目は，(10＋20)(30＋40)(50＋60)のように，前後のカッコに乗算の省略が行われている場合。

プログラムはこのそれぞれの処理に対応させるために３つ用意します。

M：それが変換テーブルの処理番号02，03，04ですね。

純：そのとおり。それぞれ１番目には03，２番目には02，３番目には04が対応しています。

Yo：ただの括弧と乗算が省略された括弧はどうやって判別するの？

老：それは括弧がどこにでてくるかでわかるのじゃ。数式は定数と演算子が順番に並んでできておるじゃろう。式の解析のときに定数のところで現れたらそれはただの括弧，演算子があるべきところで現れる括弧は乗算が省略されている括弧というわけじゃな。

そして，閉じ括弧のほうは必ず演算子があるはずのところに現れるので，式のもうひとつ後ろの項を調べて判別しなければならない。「演算子，または閉じ括弧」ときた場合はただの閉じ括弧で，「定数項，または開括弧」ときた場合は乗算が省略された閉じ括弧であると判別できるのじゃ。

Yo：開括弧，閉じ括弧の両方について調べなくちゃならないのね。。

純：普通だったら乗算の省略は考える必要はないかもしれませんが，やっぱりいつも使っている式をそのまま記述できたほうが気持ちいいですからね。

電卓の使用法

純：じゃあ，最後に電卓の使用法を説明しましょう。

M：あれ，なんだか普通と逆のような気がしますねえ。

老：まあまあ，よいではないか。どちらかといえばアルゴリズムの解説がメインなんじゃから。

純：そういうこと。で，電卓で使えるコマンドは，

？……メモリの内容を表示

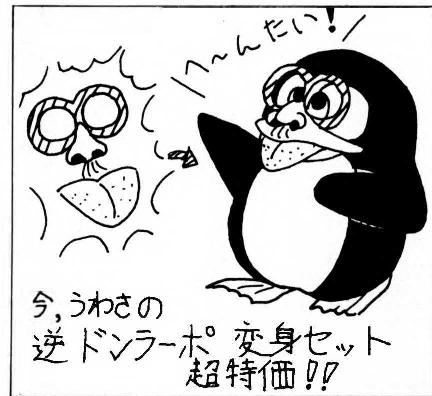
＝……Ｍ１～５のメモリに，直前に計算した答えを代入

の２つです。計算はプロンプトに続いて数式を入力すれば，答えが10進と16進で表示されます。数式で使える定数は，10進定数，16進定数（頭に＄を付ける）とメモリのＭ１～５です。使える演算子は四則演算と余算の“MOD”になっていて，単項演算子や関数はサポートしていません。と，こんなところです。

老：単項演算子や関数もサポートすれば完璧な電卓となったじゃろうに。肝心なところで手を抜きおって。

純：関数をサポートすると式のネスティングまでやらなくてはなりませんから，それは勘弁してください。

M：というところで，今日はずいぶん頑張って説明しましたね。ごくろうさんです。とりあえず前回の分のツケはこれで払ってもらいましょう。



純：わあーい。

老：じゃあ，そろそろ時間だし，わしは失礼させてもらうか。

純：あ，僕も帰ります。それじゃあ，またいつか暇になったらやってきます。さようなら。

M：毎度どうも。……と。約1名おいていっちゃったけど，どうしようか。

Yo：そのままでいいんじゃない。明日はゴミの日だし。

善：ぐうぐう……。

つづく

リスト1

```
9257
9257                    ;DENTAKU in Z80' Bar MAIN
9257                    ;       1990.5.1 by Junji
9257
9257                            ORG     SUBEND
9257
9257                    ENTRY
9257 3E 0C                      LD      A,$0C
9259 CD F4 1F                   CALL    #PRINT
925C CD E2 1F                   CALL    #MPRINT
925F 2A 2A 2A 20                DB      "*** DENTAKU in Z80'Bar ***",$0D
9263 44 45 4E 54
9267 41 4B 55 20
926B 69 6E 20 5A
926F 38 30 27 42
9273 61 72 20 2A
9277 2A 2A 0D
927A 00                         DB      00
927B                    MAIN2
927B 3E 3E                      LD      A,">"
927D CD F4 1F                   CALL    #PRINT
9280 11 8E 91                   LD      DE,LIGET
9283 CD D3 1F                   CALL    #GETL
9286 1A                         LD      A,(DE)
9287 FE 1B                      CP      $1B
9289 28 F0                      JR      Z,MAIN2
928B FE 3E                      CP      ">"
928D 20 EC                      JR      NZ,MAIN2
928F 13                         INC     DE
9290 1A                         LD      A,(DE)
9291 B7                         OR      A
9292 28 E7                      JR      Z,MAIN2
9294 FE 51                      CP      "Q"
9296 C8                         RET     Z
9297 21 7B 92                   LD      HL,MAIN2
929A E5                         PUSH    HL
929B ED 73 22 95                LD      (ERR+1),SP
929F FE 3F                      CP      "?"
92A1 28 07                      JR      Z,MEMPRT
92A3 FE 3D                      CP      "="
92A5 28 29                      JR      Z,MEMSET
92A7 18 73                      JR      @SHIKI
92A9 C9                         RET
92AA
92AA                    ;MEMORY PRINT
92AA                    MEMPRT
92AA 11 7B 91                   LD      DE,MEMDAT
92AD 06 05                      LD      B,05
92AF 0E 31                      LD      C,"1"
92B1                    MEM2
92B1 3E 4D                      LD      A,"M"
92B3 CD F4 1F                   CALL    #PRINT
92B6 79                         LD      A,C
92B7 CD F4 1F                   CALL    #PRINT
92BA 3E 3D                      LD      A,"="
92BC CD F4 1F                   CALL    #PRINT
92BF 1A                         LD      A,(DE)
92C0 6F                         LD      L,A
92C1 13                         INC     DE
92C2 1A                         LD      A,(DE)
92C3 67                         LD      H,A
92C4 13                         INC     DE
92C5 D5                         PUSH    DE
92C6 C5                         PUSH    BC
92C7 CD 0C 94                   CALL    HXDECPRT
92CA C1                         POP     BC
92CB D1                         POP     DE
92CC 0C                         INC     C
92CD 10 E2                      DJNZ    MEM2
92CF C9                         RET
92D0
92D0                    ;ANSWER TO MEMORY
92D0                    MEMSET
92D0 13                         INC     DE
92D1 1A                         LD      A,(DE)
92D2 FE 4D                      CP      "M"
92D4 C2 12 95                   JP      NZ,ERROR
92D7 13                         INC     DE
92D8 CD D1 90                   CALL    NUM10
92DB DA 12 95                   JP      C,ERROR
92DE 3E 05                      LD      A,05
92E0 BD                         CP      L
92E1 DA 12 95                   JP      C,ERROR
92E4 7D                         LD      A,L
92E5 B7                         OR      A
92E6 CA 12 95                   JP      Z,ERROR
92E9 7C                         LD      A,H
92EA B7                         OR      A
92EB C2 12 95                   JP      NZ,ERROR
92EE 3E 4D                      LD      A,"M"
92F0 CD F4 1F                   CALL    #PRINT
92F3 7D                         LD      A,L
92F4 C6 30                      ADD     A,"0"
92F6 CD F4 1F                   CALL    #PRINT
92F9 3E 3D                      LD      A,"="
92FB CD F4 1F                   CALL    #PRINT
92FE 7D                         LD      A,L
92FF 3D                         DEC     A
9300 CD 12 93                   CALL    MEMADR
9303 ED 5B 8C 91                LD      DE,(ANSWER)
9307 73                         LD      (HL),E
9308 23                         INC     HL
9309 72                         LD      (HL),D
930A EB                         EX      DE,HL
930B CD 0B 91                   CALL    STRING16
930E CD EE 1F                   CALL    #LTNL
9311 C9                         RET
9312
9312                    ;MEMORY ADRESS
9312                    MEMADR
9312 21 7B 91                   LD      HL,MEMDAT
9315                    MEMR2
9315 B7                         OR      A
9316 C8                         RET     Z
9317 23                         INC     HL
9318 23                         INC     HL
9319 3D                         DEC     A
931A 18 F9                      JR      MEMR2
931C
931C                    ;SHIKI NO TENKAI
931C                    @SHIKI
931C 21 06 92                   LD      HL,WPSP
931F 22 86 91                   LD      (WPADR),HL
9322 21 42 92                   LD      HL,ENZSP
9325 22 88 91                   LD      (ESPADR),HL
9328 CD 35 93                   CALL    SHIKI
932B 21 06 92                   LD      HL,WPSP
932E 22 86 91                   LD      (WPADR),HL
9331 CD DB 93                   CALL    CALSHIKI
9334 C9                         RET
9335
9335                    SHIKI
9335 1A                         LD      A,(DE)
9336 FE 00                      CP      $00             ;SHIKI NO END CODE
9338 20 04                      JR      NZ,SHIKI8
933A                    SHIKI7
933A 06 0A                      LD      B,$0A
933C 18 47                      JR      SHIKI6
933E                    SHIKI8
933E FE 29                      CP      ")"
9340 CA 12 95                   JP      Z,ERROR
9343 FE 28                      CP      "("
9345 20 08                      JR      NZ,SHIKI2
9347 13                         INC     DE
9348 3E 06                      LD      A,06            ;(
934A CD 2E 90                   CALL    ENZPUSH
934D 18 E6                      JR      SHIKI
934F                    SHIKI2
934F CD CA 94                   CALL    TEISU
9352 DA 12 95                   JP      C,ERROR
9355                    SHIKI3
9355 1A                         LD      A,(DE)
9356 B7                         OR      A               ;SHIKI NO END CODE
9357 28 E1                      JR      Z,SHIKI7
9359 FE 29                      CP      ")"
935B 20 1F                      JR      NZ,SHIKI4
935D 13                         INC     DE
935E 06 09                      LD      B,09            ;)
9360 1A                         LD      A,(DE)
9361 FE 00                      CP      $00             ;SHIKI NO END CODE
9363 28 20                      JR      Z,SHIKI6
9365 FE 28                      CP      "("
9367 28 10                      JR      Z,SHIKI5
9369 FE 29                      CP      ")"
936B 28 18                      JR      Z,SHIKI6
936D 21 62 91                   LD      HL,CALTBL
9370 D5                         PUSH    DE
9371 C5                         PUSH    BC
9372 CD 5A 90                   CALL    SEARCHSUB
9375 C1                         POP     BC
9376 D1                         POP     DE
9377 30 0C                      JR      NC,SHIKI6
9379                    SHIKI5
9379 05                         DEC     B               ;)*
937A 18 09                      JR      SHIKI6
937C                    SHIKI4
937C 21 62 91                   LD      HL,CALTBL
937F CD 5A 90                   CALL    SEARCHSUB
9382 DA 12 95                   JP      C,ERROR
9385                    SHIKI6
9385 CD 58 94                   CALL    TBLNUM
9388 B7                         OR      A
9389 28 12                      JR      Z,SPCHK
938B 3D                         DEC     A
938C 28 19                      JR      Z,SPPUSH
938E 3D                         DEC     A
938F 28 1D                      JR      Z,MULTSP
9391 3D                         DEC     A
9392 28 2B                      JR      Z,SPDROP
9394 3D                         DEC     A
9395 28 2E                      JR      Z,DBMULTSP
9397 3D                         DEC     A
9398 28 38                      JR      Z,COMPLETE
939A C3 12 95                   JP      ERROR
939D                    SPCHK
939D CD 37 90                   CALL    ENZPOP
93A0 C5                         PUSH    BC
93A1 CD 03 90                   CALL    OUTENZ          ;WORK NI SHUTURYOKU
93A4 C1                         POP     BC
93A5 18 DE                      JR      SHIKI6
93A7                    SPPUSH
```

```
93A7 78                  LD    A,B
93A8 CD 2E 90            CALL  ENZPUSH
93AB C3 35 93            JP    SHIKI
93AE            MULTSP
93AE CD 37 90            CALL  ENZPOP
93B1 3E 00               LD    A,$00
93B3 CD 2E 90            CALL  ENZPUSH
93B6 78                  LD    A,B
93B7 FE 09               CP    $09        ;)
93B9 CA 55 93            JP    Z,SHIKI3
93BC C3 35 93            JP    SHIKI
93BF            SPDROP
93BF CD 37 90            CALL  ENZPOP
93C2 C3 55 93            JP    SHIKI3
93C5
93C5            DBMULTSP
93C5 CD 37 90            CALL  ENZPOP
93C8 97                  SUB   A
93C9 CD 2E 90            CALL  ENZPUSH
93CC CD 2E 90            CALL  ENZPUSH
93CF C3 35 93            JP    SHIKI
93D2
93D2            ;SHIKI END
93D2            COMPLETE
93D2 3E FF               LD    A,$FF      ;WORK TO END CODE
93D4 01 FF FF            LD    BC,$FFFF
93D7 CD 40 90            CALL  WPWRITE
93DA C9                  RET
93DB
93DB            ;KEISAN MAIN
93DB            CALSHIKI
93DB 21 06 92            LD    HL,CALSP
93DE 22 8A 91            LD    (CALADR),HL
93E1            CAL4
93E1 CD 4D 90            CALL  WPREAD
93E4 FE FF               CP    $FF
93E6 28 05               JR    Z,CAL2
93E8 CD 21 94            CALL  CALMAIN
93EB 18 F4               JR    CAL4
93ED            CAL2
93ED ED 73 FA 93         LD    (CAL3+1),SP
93F1 ED 7B 8A 91         LD    SP,(CALADR)
93F5 E1                  POP   HL         ;ANSWER
93F6 22 8C 91            LD    (ANSWER),HL
93F9 31 00 00   CAL3     LD    SP,0000
93FC CD E2 1F            CALL  #MPRINT
93FF 41 4E 53 57         DB    "ANSWER- ",00
9403 45 52 3D 20
9407 00
9408 CD 0C 94            CALL  HXDECPRT
940B C9                  RET
940C
940C            HXDECPRT
940C E5                  PUSH  HL
940D CD 0B 91            CALL  STRING16
9410 3E 28               LD    A,"("
9412 CD F4 1F            CALL  #PRINT
9415 E1                  POP   HL
9416 CD BE 1F            CALL  #PRTHL
9419 CD E2 1F            CALL  #MPRINT
941C 48 29 0D 00         DB    "H)",$0D,00
9420 C9                  RET
9421
9421            CALMAIN
9421 B7                  OR    A
9422 28 07               JR    Z,PTEISU
9424 3D                  DEC   A
9425 28 08               JR    Z,PMEMORY
9427 3D                  DEC   A
9428 28 0E               JR    Z,ENZAN
942A C9                  RET
942B            PTEISU
942B CD 0C 90            CALL  PUSHD
942E C9                  RET
942F            PMEMORY
942F 79                  LD    A,C
9430 CD 12 93            CALL  MEMADR
9433 4E                  LD    C,(HL)
9434 23                  INC   HL
9435 46                  LD    B,(HL)
9436 18 F3               JR    PTEISU
9438            ENZAN
9438 21 52 94            LD    HL,EZ2
943B E5                  PUSH  HL
943C 60                  LD    H,B
943D 69                  LD    L,C
943E 29                  ADD   HL,HL
943F 11 71 91            LD    DE,JUMPTBL
9442 19                  ADD   HL,DE
9443 4E                  LD    C,(HL)
9444 23                  INC   HL
9445 46                  LD    B,(HL)
9446
9446 C5                  PUSH  BC
9447 CD 1D 90            CALL  POPD
944A 59                  LD    E,C
944B 50                  LD    D,B
944C CD 1D 90            CALL  POPD
944F 69                  LD    L,C
9450 60                  LD    H,B
9451 C9                  RET
9452            EZ2
9452 4D                  LD    C,L
9453 44                  LD    B,H
9454 CD 0C 90            CALL  PUSHD
9457 C9                  RET
9458
9458            ;TABLE NUMBER GET
9458            TBLNUM
9458 D5                  PUSH  DE
9459 21 72 94            LD    HL,ENZTBL
945C C5                  PUSH  BC
945D 78                  LD    A,B
945E
945E B7                  OR    A
945F 11 08 00            LD    DE,08
9462 28 03               JR    Z,TBNUM2
9464            TBNUM3
9464 19                  ADD   HL,DE
9465 10 FD               DJNZ  TBNUM3
9467            TBNUM2
9467 C1                  POP   BC
9468 E5                  PUSH  HL
9469
9469 2A 88 91            LD    HL,(ESPADR)
946C 5E                  LD    E,(HL)
946D E1                  POP   HL
946E 19                  ADD   HL,DE
946F 7E                  LD    A,(HL)
9470 D1                  POP   DE
9471 C9                  RET
9472
9472            ENZTBL   ;     *  / MOD + - *( ( SN
9472 00 00 00 01         DB    00,00,00,01,01,01,01,01  ;*
9476 01 01 01 01
947A 00 00 00 01         DB    00,00,00,01,01,01,01,01  ;/
947E 01 01 01 01
9482 00 00 00 01         DB    00,00,00,01,01,01,01,01  ;MOD
9486 01 01 01 01
948A 00 00 00 00         DB    00,00,00,00,00,01,01,01  ;+
948E 00 01 01 01
9492 00 00 00 00         DB    00,00,00,00,00,01,01,01  ;-
9496 00 01 01 01
949A
949A 01 01 01 01         DB    01,01,01,01,01,01,01,01  ;*(
949E 01 01 01 01
94A2 01 01 01 01         DB    01,01,01,01,01,01,01,01  ;(
94A6 01 01 01 01
94AA 01 01 01 01         DB    01,01,01,01,01,01,01,01  ;DUMMY
94AE 01 01 01 01
94B2 00 00 00 00         DB    00,00,00,00,00,04,02,99  ;)*
94B6 00 04 02 63
94BA 00 00 00 00         DB    00,00,00,00,00,02,03,99  ;)
94BE 00 02 03 63
94C2 00 00 00 00         DB    00,00,00,00,00,99,99,05  ;SP END
94C6 00 63 63 05
94CA
94CA            ;TEISU NO HANTEI
94CA            TEISU
94CA CD 04 95            CALL  TEICHK
94CD D8                  RET   C
94CE 1A                  LD    A,(DE)
94CF FE 4D               CP    "M"
94D1 28 0D               JR    Z,TEISU2
94D3 FE 24               CP    "$"
94D5 28 1B               JR    Z,HEXTEI
94D7 CD D1 90            CALL  NUM10
94DA D8                  RET   C
94DB            DECTEI
94DB 97                  SUB   A
94DC 4D                  LD    C,L
94DD 44                  LD    B,H
94DE 18 0E               JR    TEISU3
94E0            TEISU2
94E0 47                  LD    B,A
94E1 13                  INC   DE
94E2 1A                  LD    A,(DE)
94E3 D6 31               SUB   "1"
94E5 D8                  RET   C
94E6 FE 04               CP    04
94E8 3F                  CCF
94E9 D8                  RET   C
94EA 13                  INC   DE
94EB 4F                  LD    C,A
94EC 3E 01               LD    A,01
94EE            TEISU3
94EE CD 40 90            CALL  WPWRITE
94F1 C9                  RET
94F2            HEXTEI
94F2 13                  INC   DE
94F3 D5                  PUSH  DE
94F4 CD B2 1F            CALL  #HLHEX
94F7 C1                  POP   BC
94F8 30 E1               JR    NC,DECTEI
94FA 59                  LD    E,C
94FB 50                  LD    D,B
94FC CD B5 1F            CALL  #2HEX
94FF 26 00               LD    H,00
9501 6F                  LD    L,A
9502 18 D7               JR    DECTEI
9504
9504            TEICHK
9504 1A                  LD    A,(DE)
9505 FE 4D               CP    "M"
9507 C8                  RET   Z
9508 FE 24               CP    "$"
950A C8                  RET   Z
950B FE 30               CP    "0"
950D D8                  RET   C
950E FE 3A               CP    "9"+1
9510 3F                  CCF
9511 C9                  RET
9512
9512            ;ERROR PRINT
9512            ERROR
9512 CD EE 1F            CALL  #LTNL
9515 CD E2 1F            CALL  #MPRINT
9518 45 52 52 4F         DB    "ERROR!!",$0D
951C 52 21 21 0D
9520 00                  DB    00
9521 31 00 00   ERR      LD    SP,0000
9524 C9                  RET
```

★(で)のショートプロぱーてぃ その12

祝! 1周年記念

Komura Satoshi 古村　聡

今回こそは本当に1周年記念だよー。げに恐ろしきは勘違いかな。さて，今回のショートプロはどこかで見たことがあるようなX1用ゲーム「THE FANFAN」とちょっと変わったX68000用「かべくずし」です。おまけの企画もあるよ。



illustration : T.Takahashi



1周年のごあいさつ

どーもっ！　いきなり原稿が落ちてしまったり，なぜか1周年の前夜祭を開いてしまったりといろいろアクシデントもありました。が，ついにこのショートプロぱーてぃも本当の1周年を迎えることができました。めでたいめでたい！　ということで特別企画として囲みを用意しましたのでぜひ読んでくださいね。

いやぁ，それにしてもこのショートプロの企画が出たときは「とりあえず3カ月がんばってね」ということだったんで，まさかショートプロ1周年，さらにハンズ延長戦突入（ハンズを読んでね）というところまで続くとは思ってもなかったんですよね。これもひとえに，いつも楽しいイラストを描いてくれる高橋哲史くん，毎月のように破られる締め切りに「おい，明日は原稿持ってくるんやろな」とドスのきいた関西弁で励ましてくれる編集担当様，いつもひとをオモチャにして遊んでくれるスタッフのみんな，そしてやっぱり，プログラムやらハガキやら毒物飲料40本入りの段ボール箱やらでいろいろと連載にネタを提供してくれる読者の皆さんのおかげなのです。本当に本当にありがとう。これからも私のことなんでいろいろとアクシデントもあろうかとは思いますが，これからも見捨てないでショートプロぱーてぃを読んでやってください。よろしくお願いします。

以上，(で)からの1周年のごあいさつでした。





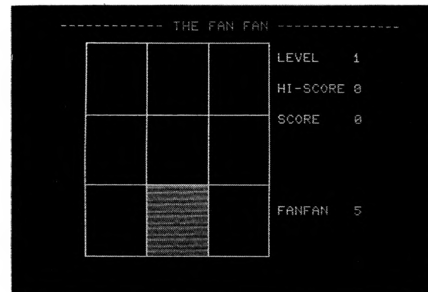
ピポパで勝負!

さーてと，そろそろいつものショートプロをいきますか。今月の1本目は「自信があります」(おおっ)と言い切ってくれた遠藤さんの反射型アクションゲーム「THE FANFAN」です。

THE FANFAN　for　X1シリーズ
(CZ-8FB01)
栃木県　遠藤亮司

プログラムをRUNさせると画面にマス目が描かれます。実はこのマス目はテンキーの1から9までのキーに対応しているんです。スペースキーを押すとマスが「ぱっ，ぱっ」と赤く点滅していきますから，しっかりとその順番を覚えておいてください。あ，動き終わりましたね。次はプレイヤーの番です。ピポパとマスが赤く光ったのと同じ順番で対応する1から9のキーを押してやってください。ポポポ。点滅したのと同じ順番でうまくテンキーを押すことができれば1面クリア。とーぜんですけど，面が進むごとに(2面ごとにだけどね)「ピポパ」の数が増えて，だんだん難しくなっていきます。テンキーを押すのを5回失敗してしまうとゲームオーバーです。

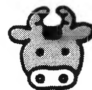
……でぇーい，ちゃかちゃか，ちゃかちゃか，動くんじゃねえやい。覚え切れないじゃないか！　結構なスピードでマスが点滅していくので順番をちっとも憶えられないんですよねー。うーん，なかなかに瞬間的な記憶力を要するゲームです。こりゃー確かに反射型アクションゲームだ。いや，正確には「反射的な記憶力をつけるゲーム」かもしれない。なんかこのゲームをやったあとって爽快な感じになれますね。もっとも，この手のゲームってパズルゲームの次に神経衰弱が苦手な私としては（苦手なものの多い男だな）結構つらいゲームでもあったりするんですけどね。



THE FANFAN

ん，なになに，「仮面ノリダーのファンファン大佐が使っていたモニタディスプレイからこのゲームを思いつきました。よって，THE FANFANであります」ってか。なるほどねー。ふとゲームのネタを思いついたっていうのはよく聞く話ですけど，テレビからネタを拾ってくるというのは結構いい手かもしれませんね。テレビにしてもゲームにしても人を楽しませるものであることには変わりないわけですから。やっぱり人を楽しませようと思ったら，エンターテイメントの先輩であるテレビやマンガを見習うというのは結構いい方法じゃないかなーと思ったりします。

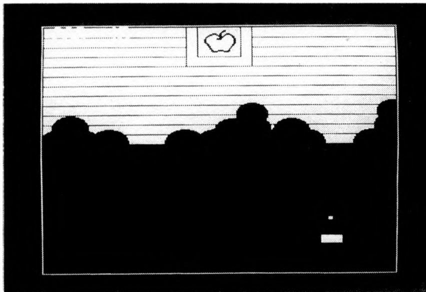
もっとも，安易にテレビからキャラクターを借りてきてクソゲーなんか作っちゃうと大ひんしゅくを買いかねないわけだけど（しかし，これもよくある話）。



縮めて縮めたかべくずし

さて，続いて今月の2本目です。

かべくずし　for　X68000
(X-BASIC)
東京都　太田敬三


かべくずし

パドルを動かしてタマに当てて，画面上半分の壁をくずし，画面中央の絵を破壊する。そう，早い話がブロックくずしなんです。なーんだと思ったそこのあなた，あなたは甘い，甘すぎるっ！　つい力が入ってしまいましたが，そこはショートプロに載るプログラム，当然ただのブロックくずしではありません。

まず，パドルはジョイスティックで８方向に動かすことができます。そう，左右だけに動くのではないんです。んで，さらにジョイスティックのボタンを押しながらだと速く動いて，そーれそらしたタマと追いかけっこだ，てなもんです(ちなみにボタンを２つとも押しちゃうと，ほとんど操作不能なくらい速くなる)。ほらねっ，普通のとはちょっとばかし違うでしょ。

うーん，それはともかく，タマが壁にぶつかったときがいいですねえ。普通みたいに当たった１ブロックが消えるんじゃなくて「ババババッ」と爆発して，まるーくえぐれるんですね。そして，ブロックくずしのように玉がブロックとブロックの間で往復運動したりもする(この表現でわかるかな？)わけなんですが，このときなんか爆発してるのがとーっても綺麗です。そうそう，ターボボタン（ジョイスティックのトリガーを押すとパドルの移動が速くなる）機能は私が勝手につけてしまいました。作者の太田さん，ごめんなさい。

ちなみにどうやったかというと，パドルの移動ルーチン(360,370行)のところで，strig(1)関数で取ってきた値をパドルの移動の増分値にかけてるだけです(だから，ボタンの左右どちらが押されたかで速さがちがう。そのうえ，同時に押すと上記のようにメチャ速くなるわけ)。

いやそれにしても，ゲーム自体もたいしたものだけど，それ以上によくプログラムを小さくまとめたなーと感心してしまいました。特に，破壊目標である絵。よくこの小さいプログラムでこんな絵を表示させましたねー。ええ，この短いプログラムのどこにそんな絵のパターン（しかも毎回毎回

リスト1 THE FANFAN

```
10 CLS4:WIDTH 40:INIT:DIMA(7,7),B(7,7):PRW 254:HI=0:REPEAT OFF
20 LINE(24,16)-(192,183),PSET,B
30 LINE(80,16)-(136,183),PSET,B
40 LINE(24,72)-(192,127),PSET,B
50 FORI=1 TO7:COLOR2:PRINT"[redacted]":NEXT
60 GET@(0,0)-(7,7),A:CLS
70 GET@(0,0)-(7,7),B
80 COLOR4:PRINT"------------ THE FAN FAN ---------------":COLOR5
90 N=3:F=0:F1=1:LE=1:SC=0:OV=5
100 GOSUB540
110 LOCATE 25,21:PRINT"HIT SPACE KEY"
120 IF INKEY$=" " THEN130 ELSE120
130 LOCATE 25,21:PRINT"              "
140 FOR L=0 TO N
150 XX=INT(RND*3):YY=INT(RND*3)
160 IF XX=0 THEN X=3
170 IF XX=1 THEN X=10
180 IF XX=2 THEN X=17
190 IF YY=0 THEN Y=2
200 IF YY=1 THEN Y=9
210 IF YY=2 THEN Y=16
220 PUT@(X,Y)-(X+7,Y+7),A:PLAY"L0GFG"
230 PUT@(X,Y)-(X+7,Y+7),B
240 IF F=1 THEN RETURN
250 IF X=3 ANDY=2  THEN K=7
260 IF X=3 ANDY=9  THEN K=4
270 IF X=3 ANDY=16 THEN K=1
280 IF X=10ANDY=2  THEN K=8
290 IF X=10ANDY=9  THEN K=5
300 IF X=10ANDY=16 THEN K=2
310 IF X=17ANDY=2  THEN K=9
320 IF X=17ANDY=9  THEN K=6
330 IF X=17ANDY=16 THEN K=3
340 POKE &HC000+L,K
350 NEXT
360 FOR L=0 TO N
370 K$=INPUT$(1)
380 K1=PEEK(&HC000+L):F=1
390 IF K$="1" THEN X=3 :Y=16:K2=1
400 IF K$="2" THEN X=10:Y=16:K2=2
410 IF K$="3" THEN X=17:Y=16:K2=3
420 IF K$="4" THEN X=3 :Y=9 :K2=4
430 IF K$="5" THEN X=10:Y=9 :K2=5
440 IF K$="6" THEN X=17:Y=9 :K2=6
450 IF K$="7" THEN X=3 :Y=2 :K2=7
460 IF K$="8" THEN X=10:Y=2 :K2=8
470 IF K$="9" THEN X=17:Y=2 :K2=9
480 IF K2=K1 THENSC=SC+5:GOSUB220 ELSE PLAY"DCD":L=L-1:OV=OV-1
490 GOSUB540
500 IF OV=0 THEN600
510 NEXT:F=0:F1=F1+1
520 IF F1>2 THEN LE=LE+1:N=N+1:F1=1
530 GOTO100
540 IF HI<SC THEN HI=SC
550 LOCATE25,3:PRINT"LEVEL    ";LE
560 LOCATE25,6:PRINT"HI-SCORE";HI
570 LOCATE25,9:PRINT"SCORE    ";SC
580 LOCATE25,18:PRINT"FANFAN   ";OV
590 RETURN
600 LOCATE25,21:PRINT"++GAME OVER++"
610 FOR T=0 TO 5000:NEXT:CLS:GOTO80
```

リスト2 かべくずし

```
10 int a,b,ch,ta,k,x,y,m,n,v,w,i,j,sc,sa,hs,sg:dim char z(255)
20 color 3:screen 0,1,1,1:apage(1):box(0,0,255,255,1):apage(0)
30 sp_init():sp_disp(1):sp_on(0,1):m_alloc(1,10)
40 for a=0 to 2:for b=0 to 2:z(a*16+b)=14-(b=1):next:next:sp_def(1,z)
50 for a=0 to 7:for b=0 to 14:z(a*16+b)=14-(a=2)-(a=3):next:next:sp_def(0,z)
60 randomize(543*atoi(right$(time$,2))):ch=int(rnd()*19)+165
70 while 1
80   color 7:sc=0:ta=0
90   while 1
100    start():if game() then break
110    for a=0 to 9999:next
120    ta=ta+20:if ta>120 then sc=sc+50000:ta=0
130  endwhile
140  for a=0 to 127:box(a,a,255-a,255-a,0):next
150  locate 0,1
160  if sc>=hs then hs=sc:print"こりゃあすごいなハイスコアだよ！" else {
170    print"めざすは";hs;"がんばろう！" }
180  color 3:repeat:until strig(1)
190 endwhile
200 end
210 func start()
220 sa=10:v=int(rnd()*224)+16:w=248:i=0:j=0:x=v:y=246:m=4*int(rnd()*2)-2:n=-3
230 ch=ch+1:if ch=184 then ch=165
240 wipe():cls:locate 0,0:print sc
250 a=int(rnd()*192)
```

絵が違う）があるんだ？　と，思わず探してしまったじゃないですか。ふーん，そうか。X68000にはこういう絵のパターンがちゃんとあって，そうやると絵が出せるんですか。私は知りませんでした。ねえねえ，太田さん，どこでこんな方法を知ったのかこっそり教えてくれません？（←外字定義をしたことのないやつ）

ただ，このゲームよくできているんだけど，壁に当たったときパドルの動きが止まっちゃうのと，あとリスト中に全然注釈がないのがちょっと残念なんだよねー。これから投稿する人はぜひプログラムに注釈をつけてくださいね。

うーむ，それにしてもこのコーナーはいつまで続けられるのかなー。とりあえずマシン語カクテルとはスタートがほとんど同じなので負けたくないなー。そんなこんなでまた来月。

```
260 for b=2 to 15:palet(b,hsv(a,31,31)):a=a+4:a=a+192*(a>191):next
270 a=a+66:a=a+192*(a>191):palet(1,hsv(a,31,31))
280 for a=0 to 11:fill(2,a*10+2+ta,253,a*10+10+ta,a+3):next
290 box(104,1,151,41,0):fill(105,2,150,40,2)
300 box(112,1,143,31,0):fill(113,2,142,30,15)
310 symbol(116,3,chr$(235)+chr$(ch),1,1,2,0,0)
320 sp_move(0,v-7,w,0):sp_move(1,x-1,y-1,1):repeat:until stick(1)
330 endfunc
340 func game()
350 k=stick(1):sg=strig(1)
360 i=(3+sg*3)*((k=1)+(k=4)+(k=7)-(k=3)-(k=6)-(k=9)):v=v+i
370 if v and 256 then v=v-i
380 j=(k>6)-(k<4)+(k=0) shl (2+sg):w=w+j:if not w and 128 then w=w-j
390 x=x+m:if x and 256 then m=-m:x=x+m
400 if point(x,y) then m=-m:if dokan(point(x,y)) then return(0)
410 y=y+n:if y and 256 then if y<0 then n=3:y=0 else return(1)
420 if point(x,y) then n=-n:if dokan(point(x,y)) then return(0)
430 sp_move(1,x-1,y-1,1):sp_move(0,v-7,w,0)
440 if abs(v-x)<9 and abs(w-y)<4 then pakon()
450 goto 350
460 endfunc
470 func dokan(a)
480 for b=1 to a:circle(x,y,b,1):next
490 m_init()
500 if a=15 then sa=sa+11111:m_trk(1,"V15@6801C") else m_trk(1,"V10@6802C")
510 m_play():sc=sc+sa:locate 0,0:print sc;sa;chr$(5):sa=sa+11
520 for b=1 to a:circle(x,y+1,b,0):circle(x,y,b,0):next
530 return(a=15)
540 endfunc
550 func pakon()
560 if n<0 then return()
570 m_init():m_trk(1,"V10@6706E"):m_play()
580 sa=10:n=-3:if abs(i+m)=1 then m=-m
590 endfunc
```

１周年特別企画――どんちゃん騒ぎの部屋

えー，ささやかながら１周年特別企画として（手前ミソくさくてちょっと恥ずかしいんだけどね）皆様からショートプロぱーてぃに寄せられたご意見，ご感想，文句に苦情，祝辞の言葉などなどにお答えしたいと思います。最初の方，どーぞ！

☆ほう。古村氏の連載も１周年ですか。月日は百代の過客にして天上天下唯我独尊。で，人間には３種類あると仮定しよう。(1)普段は無口だが，文章を書かせるととても面白くて含蓄深いことを書く奴，(2)普段のノリがそのまま文章に出る（それ以外は書けなかったりする）ヤツ，(3)普段は面白いのに，文章はまったくつまらないやつ，である。(で)君はといえば，いわずもがな(2)である。彼はあのとおりの人格なのだから。というわけで，変にウケ狙いなどせず，すくすく伸びて，成長した姿を読ませてほしい。それが非常に楽しみである(爆笑)。（荻窪圭）

へへーっ！　荻窪師匠からの祝辞だー！　いつもお世話になってますう。そーです，私はそれ以外は書けないんです。ちなみに荻窪師匠は(4)書いている文章もすごいが本物に会ってみるとさらにすごいので恐れ入ってしまうタイプ。つまり人間がはるかに深い(人物から文章の想像はつくけど文章から人物を想像できない)っていうことで尊敬しています。はい。今度飲みにいきましょうよ，師匠。もちろん荻窪師匠のおごりでね！　しかし，悪友金子俊一とかグラフィックの魔術師丹明彦さんにも祝辞を頼んだのになー。いったいどーなってるんだろ。

☆（で）さんの初登場は(ピー)年(ピー)月号の(ピー)のレビューではないですか？　違っていたらすいません。

（アンケートハガキより，原正人さん）

ピンポンピンポーン！　大正解です。えー，あの頃は（で）って使ってなかったのによくわかりましたねー。まだ，Oh!X編集部がいまの泉岳寺に移るまえのまえ，四番町の半地下の編集部の頃の話だからねー。懐かしいなあ。ちなみに本文中の「ピー」は私がつけたものですが，別に恐ろしいことが書いてあるわけではありません。あしからず。

☆５月号の「空飛ぶDNAデモ」を走らせてみた。それを見た友人曰く，「まんが日本昔ばなし」のオープニングみたいだと。

（アンケートハガキより，神生直敏さん）

おー。「ぼうやー，よい子だ～♪」というあれですね(そういえばパロディで「ぼうやーよい子だ金だしなー」というのがあったな)。しかし，あのデモは本当に好評でした。そうそう，某MS-DOSマシンにも似たようなデモがあるという話を聞いたのである人に見せてもらったのですが，見た瞬間，「勝った！」と思ってしまいました。

☆ちょーどゲームでも作ってみようと思ってたとこなんですよ，（で）さん。シューティングじゃないけど。４月号の外部関数は役に立ちそうです。（アンケートハガキより，小林到さん）

わーい，それはよかった。うれしいです。ぱーてぃハンズはなかなかに評判がよいので喜んでおります。それに４月号のsp_chk( )も５月号のデモに負けず劣らず好評でした。ここんとこいい投稿が多い。ゆえにショートプロの評判も上がるというわけでとてもうれしい。小林さんもぜひ投稿してみてくださいね。

☆ライターのプロフィールが知りたい。

（アンケートハガキより，桐山秀幸さん）

ショートプロとは関係ないけど，思わず持ってきてしまいました。あははは，私も知りたい。うちの編集部は謎の人物がいっぱいいるから壮絶なものになること間違いなし。でも，自分のプロフィールは遠慮したい……。

☆すいませーん。Reserved featureエラーがでるんですけどー。（バグ電話より）

すいませーん。X１のBASIC（CZ-8FB01）には新旧のBASIC（ver.１と２）があるのはご存じですよね。ショートプロのものはほとんどがどちらのBASICでも使えるのですが，５月号のDIG MANは旧BASIC専用だったんですよねー。うっかり私が書き忘れてたんです。本当にごめんなさい。今後のために（やらないように心がけるつもりではありますけど）一応，こういうときの対処の方法を教えておきましょうね。とりあえず，バージョンの違うBASICで打ってしまったらASCIIセーブしてください。

SAVE “ファイル名”, A

それからリセットして本来使うはずだったBASICを立ち上げます。そして，再びロードすればOKです。

おお，そうだ。ショートプロで質問の多かったものに５月号のDNAデモがあるんですが，これはコンパイル時のスイッチを小文字にしてしまった人が多かったみたいです。コンパイルできなかった人はそこを注意してもう一度やってみてください。リストにバグはありません。

うーむ，なにやら「あの筋？」質問箱になってしまった。

☆（で）のぱーてぃハンズ（その３）はものすごーくうれしい。

（アンケートハガキ，白井達広さん）

ありがとー。私も本当にうれしいです。あのハンズって結構大変なんですよ。なにしろOh!Xには珍しく毎月ちょっとずつプログラムを載せていく形式なんで始める前の下準備がめんどくさいわ，文字が小さいから１ページでショートプロ３ページ分の文章を書かなくちゃいけなかったりするわなんですよね。その努力が報われたわけで，いや，よかったよかった。延長戦もよろしくね（あと，リクエストもね）。

うーん，アンケートハガキっていいなあ。と思ってるとこんなハガキもあったりします。

☆（で）のぱーてぃハンズのコーナーを３ページくらいに増やしてほしい。

（アンケートハガキ，箕浦健一郎さん）

……かんべんしてよ（でもうれしい！）。ま，なにはともあれ，これからもよろしくお願いします。

(で)のぱーてぃハンズ―――――（その5）

恵まれている（で）に愛の手を！

さあて，さてさて。結構のんびりやっていたはずのこのコーナーもいよいよ今月の敵と敵のタマの動き，そして来月の当たり判定を残すのみになってしまったんです。ということで本来なら来月で「それではみなさん，さよーならー」となるはずだったのですが，皆様のハガキのおかげで再来月からぱーてぃハンズは第2部に突入することとなりました。はい，拍手拍手！　でも，連載が延長になるのはうれしいんだけど，まさかこうなるとは予想すらしてなかったんで，はっきりいってまだなにをどうするのか全然準備してなかったりするのですよねー。困ったなー，急になんか作れっていわれてもなに作っていいのかわからんよー。てなわけでこんなものを作ってほしいとかこうしてみてはどうかとか，こういうところがわからなかったとかいうハガキを大募集しちゃいます。ネタのない（で）にあいの手をーっ！　あーこりゃこりゃ（そのあいの手じゃなーい）。

敵襲だーっ！　ゲームの個性だーっ！

さて。というわけで，敵の出現，敵の移動，敵のタマ撃ち，敵のタマの移動です。

シューティングってゲームセンターにもいろいろなものがありますけど，基本的には自機を動かしてタマを撃つという意味でそんなに変わらないですよね。シューティングゲームの個性って敵の出現，動きのパターンや背景なんかがかなりの部分を占めていると思うんです（例外も多々ありますが）。だからシューティングゲームを作るとき，背景をカラフルにしたり，デカキャラを作ったり（X68000だったら簡単でしょ），敵キャラの動きをなめらかにしたり，あと，敵のタマが多くなりすぎてバランスが悪くなってしまわないようにとかの努力をすれば市販ゲームぐらいにできなくはないと思うんですよね。特に，X68000みたいにスプライトやBGがあったりなどなどといろいろ機能が揃ってるマシンだとアフターバーナーみたいに特別なプログラムテクニックが必要なものでない限り，アマチュアの作ったゲームと売ってるゲームの差は，極端にいうとどれだけデータを作れるか（どれだけの人がどれだけの時間をかけたか），どれだけ妥協しないで作ったかの差ではないかと思います。ゲーム作りの極意は根性（もちろん創意工夫も）なんです。決してテクニックだけではありません。

皆さんにはそのようにがんばっていただきたいなー，ということで今回私は手を抜かせていただきます（い，いままで並べたゴタクはいったいなんだったんだ……）。

で，敵の動きなんですがとりあえずこんなのを考えてみました。

「敵がすすーっと下りてくる」
「ばっ，とタマをまき散らす」
「敵はすーっと逃げていく」

なんかとんでもなくいやな性格してる敵キャラですけどねー。んで，敵をとりあえず動かしてみたいんですけど，その前にちょっと思い出さなきゃいけないことがある。そうそう，先月いったあれなんです。自機も敵も同時に動かさなくちゃいけないので，かわりばんこで動かすように組んでやらなくちゃいけないんですよね。先月，自機とタマを交互に動かすために自機のメインルーチンの中に，

firemove( )

って1行入れて自分のタマを動かすルーチンを呼び出してましたよね。それと同じように，

enemy_move( )

って1行入れて敵を動かすルーチンを呼び出してやるんです。ちなみに敵のタマを動かしてやるルーチンが，

bomb_move( )

なんですけど，敵のタマももちろん同時に動くわけですよね。だからbomb_move( )もそこに入れていい……んですが，なぜかbomb_moveはenemy_moveが呼び出しています。別にこれは意味はないんです，っていうか実はなんでこうしたのかよく憶えてないんです（こらこら）。たぶん敵が動くルーチンと敵のタマを動かすルーチンだけほかのルーチンと別の日に作ったので思わずそうしてしまったんじゃないかな。別に次々とルーチンがルーチンを呼んでもかまわないっちゃかまわないんですが，やっぱりリストが読みにくいですからみなさんはちゃんとどちらかに統一しましょうね。

それはそうと敵が出てきて引き返す（折り返すっていうほうがわかりやすいかな？）ってえことは，まず，敵がどこで折り返すか決めておいて，それから敵をつつーっと下ろしていって，折り返し位置にきたら帰っていくようにすればいいわけですね。さて，ここで問題です。ここではいくつ変数を作ればいいでしょう。

自分のX座標，およびY座標
折り返し点のY座標
自分が上がっているか下がっているかのフラグ

うん，4つもあればよさそうですよね。自分が上がっているか下がっているかのフラグはたとえば，

上がっているとき＝－1
下がっているとき＝＋1

としてやれば敵を動かすときに（たとえば敵のY座標がenemy_y,フラグがenemy_sgnという名前だとしたら），

enemy_ = enemy_y＋enemy_sgn

としてやればできそうですよね。

それじゃ，1つひとつルーチンを作っていきますか。まずは敵の出現。

・enemy_appear( )

とりあえず，

「タマは出ていないか」
「自分のX座標と引き返し座標を決める」
「上がり下がりのフラグを＋1にする」

このくらいかな。で，これを敵のY座標が0の（つまり敵が現れていない）とき，このルーチンを呼んでやればいいわけね。んで，

・enemy_move( )

出てきた敵をこのルーチンで動かす。これは敵を順番に1ステップ動かすわけですね。んで折り返し点にきたら上がり下がりを1にしてタマを出させます。

・bomb_move( )

タマが出ていたらタマを1ステップ進める。で，「タマをばらまく」ことにしたわけですが，とりあえずタマは3つ出して左下，下，右下に進めます。

あー疲れた。とりあえずこんなもんかなー。さて，来月は当たりチェックやっておしまいね。んー，でも当たりチェックだけで1ページもたすの苦しそうだなー（たぶん1/4ページくらいで終わっちゃうと思うんだよねー）。ま，いいか。明日は明日の風が吹くと。来月またこのOh!Xで。ガガガガガ（と穴を掘って去る）。

```
330 enemy_move()
530 /* 敵の動き*/
540 func enemy_move()
550 for i=0 to 2
560     if enemy_y(i)>=enemy_b(i) then enemy_sgn(i)=-1:enemy_fire()
570     if enemy_y(i)>0 then enemy_y(i)=enemy_y(i)+enemy_sgn(i)*8
580    j=0:for a=0 to 2:j=j+bomb_y(i,a):next
590     if j=0 and enemy_y(i)=0 then enemy_appear()
600   sp_set(38+i,enemy_x(i),enemy_y(i),((enemy_sgn(i)-1)¥-2)*&H8000+&H123)
610 next
620 bomb_move()
630 endfunc
640 func enemy_appear()
650 enemy_b(i)=0:while (enemy_b(i)=0 or enemy_x(i)<16 or enemy_x(i)>193 )
660                 enemy_x(i)=rand()and&HF0:enemy_y(i)=8:enemy_sgn(i)=1
670                 enemy_b(i)=rand() and &HF0
680                 endwhile
690 endfunc
700 func enemy_fire():/*敵もタマを撃ったりする*/
710 for j=0 to 2
720 bomb_x(i,j)=enemy_x(i):bomb_y(i,j)=enemy_y(i)
730 next
740 endfunc
750 func bomb_move() /*タマの動き*/
760 for i=0 to 2
770 for j=0 to 2
780         if bomb_y(i,j)>0 then bm_sub()
790   next
800   next
810 endfunc
820 func bm_sub()
830 bomb_y(i,j)=bomb_y(i,j)+8
840 if j=0 then bomb_x(i,j)=bomb_x(i,j)-8:if bomb_x(i,j)<=0 or bomb_y(i,j)>256
then bomb_y(i,j)=0
850 if j=1 and bomb_y(i,j)>256  then bomb_y(i,j)=0
860 if j=2 then bomb_x(i,j)=bomb_x(i,j)+8:if bomb_x(i,j)>=192 or bomb_y(i,j)>256
 then bomb_y(i,j)=0
870   sp_set(42+i*3+j,bomb_x(i,j),bomb_y(i,j),&H122)
880 endfunc
```

X68000用 OMENS OF LOVE
X1/turbo用 ENDLESS RAIN
X68000用MUSICDRVサンプル曲 ダートフォックスより Running up！

©NAMCO

Kodama Kazuhiro
小玉 和博

Fushiki Yoshihiro
伏喜 義宏

Nishikawa Zenji
西川 善司

外は暑いようですが，皆さんいかがお過ごしでしょうか。さて，今月はT-SQUAREやXといったポピューラーソングものを2本と，MUSICDRVサンプル曲としてゲームミュージックを用意しました。ちょうど夏休みですし，打ち込んで聴いてみてください。また，100号記念としてMIDI基本テクニック特集も併設，ぜひ参考にしてください。

サンプリングは使用しておりません

X68000用に「OMENS OF LOVE」をお届けしましょう。この曲はフュージョンと呼ばれるジャンルの曲で，T-SQUAREが演奏しています。T-SQUAREは5人のグループで，カシオペアと並んで日本が世界に誇れるフュージョンバンドです。F1グランプリの曲，「TRUTH」などでおなじみですよね。

曲はインストなので，比較的FM音源だけのコンピュータでも作りやすい構成とは思いますが，テクニック命といっても過言ではないフュージョンを完全に再現するのは，かなり厳しいのではないでしょうか。特にFM音源とは相性が最悪ともいえそうなギターが前面に出ている曲は至難の技だと思います。

さて，作品のデキはといいますと，とっても気持ちいい曲になっています（？）。FM音源のみでAD PCMを使っていませんので普通のOPMDRV.Xのみで演奏できますが，サンプリングドラムに頼らなくても立派に演奏できるというお手本のような仕上がりです。もともとOPMAではボスコニアンのサンプリングデータを使用していましたので，あの元気なドラム達がそのまま使われていました。そうすると静かな曲や，落ち着いた曲などではどうしてもドラムだけが浮いてしまっていたのです。そういった意味でも，この曲ではOPMのみで演奏したほうがきれいになるのです。試しにサンプリング対応にしてみたところ，やはりドラムが浮いていました。納得できない人は自分で試してみてください。

そういえば，フェードアウトもOPMだけならきれいにキマるということもいっておきましょう。


T-SQUARE

XシリーズのX

X1用にはXの「ENDLESS RAIN」をお届けしましょう。Xはライブハウスからの叩き上げバンドです。自主制作していたアルバムが2枚あって，それがバカ売れしたためレコード会社の目にとまり，プロデビューに至ったという経歴を持っています。かなり正統派のバンドといえるでしょう。残念ながら矢板にあるSHARPさんのお抱えバンドではありません，悪しからず。

さて，作品についてですが，なかなか面白い構成をしているのではないでしょうか。ヴォーカルをギターの音でやってしまったわりには，かなりまとまりがよいといえます。前述のとおり，ギターの音は結構ムズいのです。その分を考えるとよくできているといえます。

惜しむべきこととして，曲調を考えるとちょっとドラムの音が大きいのでは？　と思えます。特にPSGのハイハットが怒鳴っています。確かに，原曲でははっきりと聞こえてはくるのですが……。ハイハットはノリを出すのにも使われますが，曲を引き締めるのにも使われます。おそらく原曲の使い方は後者でしょう。引き締めるためのハイハットが，全体的に繊細な音で構成しているのを壊してしまうのはちょっともったいないですね。



X

やはりFM音源と比べてPSGの音質が落ちてしまうのは仕方ないことですので，PSGの使い方はしっかりと考えてみましょう。ソフトウェアエンベロープを掛けてコーラスラインとか，ハイハットならボリュームを小さめにするとか，S.E.を作ってみるなどが挙げられます。ミキサーをつないでる人は，PSGの音量をFM音源の7割程度にしてみてください。あとは，好みに合わせてドラム系の音を心持ち下げて聴いてみてください。　(S.K.)

「MUSICDRV」用サンプル曲

「MUSICDRV」用のサンプル曲のプログラムとして（注意：「OPMD」では演奏できません)，ナムコのトップビュータイプのオフロードカーレースゲーム「ダートフォックス」のメインテーマ「Running up！」をお届けします。この曲は，カシオペアの「Looking up」のパロディともいえる曲で（名前まで似ていたりする)，フュージョン

風のアレンジとなっていますからそっちの筋の方にもおすすめです。なんといってもウリはチョッパーの効いたベースと耳に突き刺すような高音のシンセソロ，右左にパンするバッキングです。作曲はもちろん(？)「メタルホーク」のめがてん細江氏です。

演奏方法

対応楽器はM1/R/Tシリーズ（以下M1）専用です。M1とMT-32の両方をお持ちの方はそのシステムに対応します(MT-32のみでは演奏できません)。また，FM音源も使用しているため，ミキサーなどでミキシングしてお楽しみください。まず，演奏させる前にM1側の設定をしてやります。

0）「MUSICDRV」を,

A＞MUSICDRV #180

のように組み込んでください。

1） 曲中で使用されている音色のベンドレンジを変更してください（2，4，46，48，51，71，72，75，92：後述の「MIDI基本テク特集」で説明してあります)。

2） M1リズムキット（音色番号09，49）をEDIT PROG，F4-3「VDA1 KBD TRK」のパラメータを図Aのように設定します（リズムの音色をほかの音色よりも音量をやや大きくするため)。

3） GLOBALモードにしてDRUM KIT3の「TOM2」をすべて「TOM1」に，「OPEN HH」と「CLOSED HH」のPANを(9：1)に変更してください（図B）。

4） 次にシーケンサモードにして，F1-4「MIDI CH」で各トラックのMIDIチャンネルを1，11，12，13，14，15，16，10のように設定してください（図C)。

5） F4-1「TRACK PARAMETER」で各トラックのパンポットを(5：5)，(5：5)，(9：1)，(1：9)，(5：5)，(8：2)，(5：5)，(5：5)のように設定してください（図D：**プロテクトオフにしてから設定すること**)。

6） MT-32もお持ちの方は，M1のMIDI THRU端子からMT-32のMIDI IN へMIDIケーブルを接続し，MT-32の電源を入れてください。M1のみお持ちの方は特に接続する必要はありません(当たり前だな)。

7） メインプログラムを入力，または入力されたものをロード，RUNしてください。

注意：MUSICDRVは7月号のデバッグ(バグを取ること）を行ったものを使用してください。さもないと，FM音源の音色が正常に鳴りません。

テクニックの解説

特に変わったことはしていませんが，ダンパーとピッチベンドを多用しています。ピッチベンドのMMLデータはbnd( )という関数で作っています。たとえば，

bnd(“c”,12,8192,8875)

は，12個分の精度でピッチ8192(C)からピッチ8875(C＋)まで滑らかに変化させるMMLを生成します。値，用語の意味については後述の「MIDI基本テク特集」を参照してください。

ところで，M1はコントロールチェンジにパンがありません。そのため基本的にはリアルタイムに音をパンすることが不可能です。しかし，各トラックにあらかじめ適当なパンを設定しておき，MIDIチャンネル切り替えコマンド「@ｎ」で演奏チャンネルを切り替えることにより，パンをリアルタイムに切り替えているようなニュアンスを出すことができます。初心者のM1ユーザーは参考にしてください。（善）

図A　リズムキットの設定1

```
PROG I49  VDA1 KBD TRK   Center Key
C-1 A+00 EGtime=0 AT:0 DT:0 ST:0 RT:0
```

図B　リズムキットの設定2

```
DRUM KIT3          Closed HH1
#05  11  F#1  +009 L-57 D+00 9:1
```

図C　MIDIチャンネルの設定

```
SONG1      MIDI CH
1G  11  12  13  14  15  16  10
```

図D　パンポットの設定

```
SONGI     TRACK PARAMETRE
Tr 1  I01  V99 T+00 D+00 5:5 Prot:OFF
```

Oh!X通巻100号記念 MIDI基本テク特集

私が，サングラスをかけるとほとんどチンピラの西川善司です。6月号の創刊8周年記念のディスクに付いてきた「OPMD.X」と「MUSICDRV.X（サン・ミュージカル・サービス)」，ともに好評だったようです。両ツールともに，MIDI楽器をFM音源感覚のMMLで演奏可能なため，MML派の人間にとってはまさにたなからボタもちでしょう。

しかし，MIDI楽器はMIDI楽器。細かな表現をするのに大変重宝していた「Yコマンド」が使えないのをはじめとして，MIDI楽器はFM音源やPSGとは違った箇所が多くあります。そこで，この場を借りてMIDI楽器を使うにあたっての基本テクを，音楽特集でもないのにバババーンと公開してしまいましょう。

MIDI楽器でディチューンをやる

FM音源（OPM）では，Y48＋チャンネル番号（0～7）でピッチ（音程）を微妙にずらした音を重ねてやることによって，コーラス効果を実現できました。MIDI楽器には，こういったピッチをずらすコマンドはないのでしょうか。「ピッチ」という言葉でピンときた読者もおられるでしょう。そうです，「ピッチベンド」のコマンドを用いるのです。

「MUSICDRV」では@B8192がピッチの基準値ですので，1チャンネルはこの基準値で鳴らしてやり，もう1チャンネルは基準値@B8192±50～100程度で鳴らしてやります(FM音源部もこの方法でコーラス効果を実現してやることができます)。

「OPMD」では基準値は128ですので1チャンネルは「Y9，128」で鳴らし，もう1チャンネルは「Y9，128±1～3」程度で鳴らしてやりましょう。

ここで注意がひとつ。楽器によってベンドの範囲が違うという点です。ROLAND MT-32は初期状態で1オクターブ範囲のピッチベンドが可能ですが，KORG M1/R/Tシリーズ（以下M1）では初期状態では半音範囲です。でもご安心を。たいていの機種は，このピッチベンドの範囲についてはコンフィギュレーションが可能です。M1の場合は音色単位でこの設定が可能で，音色を呼び出したあと，「EDIT PROG」モードにし，F7-2「JOYSTICK」（図1）の「P＋02」を「P＋12」にすることにより，MT-32のような1オクターブ範囲のピッチベンドが可能となります。そうそう，パラメータを書き換えたあとはF9-1「WRITE/RENAME」（図2）で音色を再登録しなければいけませんよ。まあ，ピッチベンドは1オクターブ範囲にしておいたほうが音色の応用範囲が広くなるので，M1ユーザーはすべての音色をいまいった方法で変更しておきましょう。

ちなみに，ベンドの範囲を1オクターブにしたとき，「MUSICDRV」では半音が±683，「OPMD」では±11となります。つまり，「MUSICDRV」で，

@B8192C@B8875C　(8192＋683=8875)

図1　JOY　STICK

```
PROG I00   JOY STICK
P+00 F+00      PM00 MF0     FM00 MF0
```

図2　音色登録

```
PROG I00 A.PIANO       Write/Rename
  [ < ][ > ]        [WRITE] --> I00
```

とすると，最初の「C」はもちろん「C」ですが，2回目の「C」はC＋で発音されます。「OPMD」で同じことをするには以下のようになります。

Y9,128CY9,139C　(128＋11=139)

上の値をもっと細かいステップで与えて，各音を「&」でつないでやることにより「**ポルタメント**」を表現できます。

&のお話

読者のハガキのなかにこんなのがありました。「OPMDでC&C＋とする，とCの発音後C＋の音が同時に鳴ってしまいます」。

FM音源では上のようにすると，Cの発音後，C＋のアタック音なしに音程がC＋へと変化します。「OPMD」では「&」はキーオフしないという目印に過ぎません。ですから，次にきたC＋はCをキーオフせずに鳴ってしまいます。FM音源では1チャンネル1声という大原則があるので問題はないのですが，MIDI音源は1チャンネルで和音も発声可能なので，ハガキにあるような現象が起こるのです。

手抜きというより「OPMD」の性質上仕方のない現象なのです。しかし，後述の「ダンパー」効果を，この現象を逆手に取って実現できます。

ところで「MUSICDRV」では，ハガキにあるような例を演奏させると，

CC＋

のように「&」が削除されたようなMMLが演奏されます。つまり以前に鳴ったキーコードとは違う音が新たに発音される場合，以前に鳴っていた音は**強制的にキーオフ**されるわけです。また，**FM音源部においても同様の処理が行われる**ので注意が必要です。

では，「MUSICDRV」や「OPMD」で「タイ」や「スラー」を実現するにはどうしたらよいのでしょうか。答えは簡単。先ほど，説明した「ピッチベンド」のコマンドを使ってやればいいのです。

@B8192C&@B8875C　　MUSICDRV
Y9,128C&Y9,139C　　OPMD

おわかりいただけたでしょうか？

ダンパーってなんだー

「MUSICDRV」では「@d」というコマンドがあります。これは，「ダンパー」という機能を「オン/オフ」するものなのですが，いままでFM音源のMMLのみを使っていた人にとっては耳新しい言葉です。言葉で説明するより例を用いて説明したほうがわかりやすいので，実際に「ダンパーコマンド」を用いて楽器を演奏させてみることにしましょう。

L16R@d127CEGR2.@d0

をいま演奏させたとしましょう。「L16」はただのデフォルト音長設定，続く「R」は16分休符となります。次の「@d127」はダンパーオンのコマンドで，以後発音される音はダンパーオンの効果がかかります。最初のCが発音され続いてEが発音されますが，このときCの音はキーオフされません。同様に最後のGもCとEが鳴った状態で発音されます(つまり，この時点ではCEGの和音が鳴っている)。さて，次に付点2分休符である「R2.」がきています。通常だと無音状態となるのですが，ダンパーオンの影響で「R2.」の時間，**「CEG」の和音が鳴り続きます**。そして，やっと最後の「@d0」でダンパーオフとなり，発音されていた音はすべてキーオフされます。この例は譜面にするとちょうど図3のようになります。

「OPMD」でこれをやるには「&」を用いてやります。説明は「&」のところでしたので省きますが，上の例は「OPMD」では，

L16RC&E&G&G2.

となります。ただ「G2.」のあとにオールノートオフのメッセージを送らないと，CとEの音が鳴りっぱなしとなるので注意。

ベロシティのお話

これまた，FM音源から入ってきた人には耳新しい言葉です。MIDIの専門書などには「音の立ち上がり方の速さ」などと書いてありますが，「ボリューム」のことだと思ってくださって結構です。いい方を変えれば「どのくらいの強さで鍵盤を叩いたか」ということです。ですから，実際の音量はVコマンドの値×このベロシティの値で決定されます。「MUSICDRV」で最大の音量で演奏するには，

@v127　@u127

を最初に送ってやります。

また，リズムマシンやポータブルキーボードのなかにはVコマンドを認識しない機種があります(YAMAHA RX-8など)。そういった機種に対してはこのベロシティのみが音量を決定します。

@Lのお話

「MIDIドライバで@L1などの微小音長を多用すると遅くなります」といった内容のハガキが届きました。うーん，FM音源を酷使する人は@L1でガリガリとMMLを書く人が多いようですね。そういえば，常連の立川正之君などは8分音符以上の音長は滅多に書きません，なんていってました……。MIDIは31250bpsという，速いようで実はそんなに速くないボーレートで通信をしています。私の貧弱なMIDIシステムでも発音遅れはよくあることですから，@L1の多用でテンポが遅れるなんてことは当然といえます。こういった問題の解決策としては，

1)　@L1の使用を少し控える。
2)　内蔵FM音源に対してのみ@L1を使用する(内蔵FM音源は，MIDIで通信をしているのでなくI/OでMPUと直結しているので，かなり高速な応答が可能です)。
3)　適当なトラックのMMLの最初に「@L1R」を挿入し，割り込み周期のずれを作ってやる。

が挙げられます。

「MUSICDRV」のバグ
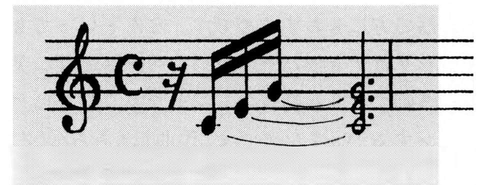
「MUSICDRV」にはバグがいくつか発見されています。「MVSET」コマンドで設定した音色番号とMMLの「@」コマンドの番号と対応しない，というバグは7月号で訂正されていますが，ほかのバグは取る手立てがないため(制作はサン・ミュージカル・サービスで，ソースリストはOh!X編集部にはありません)，これから話す解決方法で対処してください。

和音のコマンドは，

'CEG'

のように「'」内の音を同時に鳴らすもので，最初の音に音長を書けばその長さで和音が鳴り続ける

図3　ダンパーの譜面



という，いままでのMMLの常識を破った大変便利なコマンドです。しかし，

'C8.EG'

のように付点を含む音長を記述すると，暴走してしまいます。これは，デフォルト音長設定コマンド「@L」を使ってやれば簡単に同様のことができます。つまり，付点8分音符なら，

@L36 'CEG'

です。では，全音符を超えた音長で和音を鳴らし続けるにはどうしたらよいのでしょうか(「@L」では全音符である192以上は記述できません)。

答えは「ダンパー」を用いて以下のようにしてやります。

@d127 'C1EG'R1R1@d0

この例だと，全音符3個分の長さで和音CEGが鳴り続けます(原理はすでにダンパーのところで説明したのでここでは省略します)。

「@n」はMMLトラックをMIDIチャンネルいくつに割り当てるかを演奏の途中で切り替える大変便利なコマンドです。「OPMD」では「Y4,?」，「Y5,?」にあたります。「MUSICDRV」で以下のようなMMLを書いた場合，正常に動作しないので注意が必要です。

@n1@8……@n2@8……

順を追って説明すると，まず，@n1で現在演奏中のトラックをMIDIチャンネル1に変更し，次に音色切り替え「@8」で音色が切り替わります。「……」は，まあ，MML演奏データがずらーっと並んでいるとして，これらはすべてMIDIチャンネル1で音色番号8で演奏されます。

さて，次に「@n2」がきているのでトラックをMIDIチャンネル2と変更します。問題は次の「@8」で，なんとMIDIチャンネル2へ音色切り替えのメッセージが送られないのです。どうも同じ音色は切り替えないというようなアルゴリズムのもとで動作しているらしく，しかもそれを**MIDIチャンネル単位**でなくトラック単位で行っているため，このような現象が起こるのでしょう。これの対処方法としては，音色切り替え専用のトラックを設けるとか，別のトラックに音色切り替えのコマンドを挿入する，などが考えられます(ちょっと空しいね)。

いずれにせよ，サン・ミュージカル・サービスさんの迅速な対応が望まれますね。

MUSICDRVに望むこと

「MUSICDRV」はとてもよくできています。欲をいうと，ピッチベンドはオートベンドにしてほしかったし，ベンド，ベロシティやダンパー，モジュレーションなどの頻繁に使うコマンドは，できたら「@」という文字の必要のない1文字コマンドにしてほしかったです。あと和音のコマンドはFM音源にもほしかったなぁ。サン・ミュージカル・サービスさん，Ver.2に期待してます，ゴロニャーん。また，何かMUSICDRV楽器について質問があればどうぞ，Oh!X編集部・西川善司まで。

日本音楽著作権協会(出)許諾第9070779-001

リスト1 OMENS OF LOVE

```
  10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  20 /*                                                   /*
  30 /*              T h e   S Q U A R E                  /*
  40 /*                                                   /*
  50 /*           O M E N S   O F   L O V E               /*
  60 /*                                                   /*
  70 /*    P r o g r a m e d   b y   K , K o d a m a      /*
  80 /*                                                   /*
  90 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
 100 m_init():for i=1 to 8:m_alloc(i,5000):m_assign(i,i):next:m
_tempo(170)
 110 /*
 120 dim char mine(4,10)={
 130 56,15, 0, 0, 0, 0, 0, 0, 0, 3, 0,
 140 20,23, 0, 0, 1,32, 0, 1, 3, 0, 0,
 150 24,22, 0, 0, 1,16, 0, 1, 5, 0, 0,
 160 23,11, 0, 0, 0,32, 0, 1, 3, 0, 0,
 170 18, 0, 0, 6, 0, 0, 0, 1, 7, 0, 0}
 180 m_vset(70,mine)
 190 /*
 200 dim char bass(4,10)={
 210  8,15, 0, 0, 0, 0, 0, 0, 0, 3, 0,
 220 31,18, 0, 6, 2,35, 0,10, 0, 0, 0,
 230 31,14, 4, 6, 2,46, 0, 0, 7, 0, 0,
 240 31,10, 4, 6, 2,20, 1, 0, 3, 0, 0,
 250 31,10, 3, 6, 2, 0, 1, 0, 0, 0, 0}
 260 m_vset(71,bass)
 270 /*
 280 dim char eg(4,10)={
 290 58,15, 0, 0, 0, 0, 0, 0, 0, 3, 0,
 300 31, 4, 1, 1, 1,24, 0, 2, 4, 0, 0,
 310 18, 2, 1, 9, 1,20, 0, 3, 4, 0, 0,
 320 31, 4, 2, 1, 1,24, 0, 4, 1, 0, 0,
 330 31,10, 2, 7, 1, 0, 0, 1, 7, 0, 0}
 340 m_vset(72,eg)
 350 /*
 360 dim char st(4,10)={
 370 58,15, 0, 0, 0, 0, 0, 0, 0, 3, 0,
 380 31,31, 0, 5, 0,25, 1, 2, 3, 0, 0,
 390 31,31, 0, 5, 1,30, 1, 4, 1, 0, 0,
 400 31,31, 0, 4, 0,36, 1, 6, 7, 0, 0,
 410 22,31, 0, 6, 0, 0, 1, 4, 0, 0, 0}
 420 m_vset(73,st)
 430 /*
 440 dim char Sdrum(4,10)={
 450 60,15, 0, 0, 0, 0, 0, 0, 0, 3, 0,
 460 31, 0, 0, 0, 0, 5, 0, 2, 0, 0, 0,
 470 31,14,11,12,15, 0, 0, 0, 0, 0, 0,
 480 31,25, 0, 0,15, 0, 0, 1, 0, 0, 0,
 490 31,16,14,10,15, 0, 0, 1, 0, 0, 0}
 500 m_vset(75,Sdrum)
 510 /*
 520 dim char Bdrum(4,10)={
 530 60,15, 3, 0, 0, 0, 0, 0, 0, 3, 0,
 540 31,21, 0, 5,15,11, 0, 4, 0, 0, 0,
 550 29,19, 0,15,15, 0, 2, 3, 0, 0, 0,
 560 31,18, 0,15,15,21, 2, 1, 0, 0, 0,
 570 29,17, 0,15,15, 0, 0, 1, 0, 0, 0}
 580 m_vset(76,Bdrum)
 590 /*
 600 dim char elp(4,10)={
 610 50,15, 0, 0, 0, 0, 0, 0, 0, 3, 0,
 620 26, 2, 6, 6, 1,38, 2, 1, 7, 0, 0,
 630 21,10, 6, 4, 2,35, 1, 7, 3, 0, 0,
 640 24, 4, 5, 5,15,32, 0, 3, 3, 0, 0,
 650 22,15, 4, 7, 1, 0, 1, 1, 0, 0, 0}
 660 m_vset(74,elp)
 670 /*
 680 str a[256],b[256],c[256],d[256],e[256],f[256],g[256],h[256
],j[256],k[256],l[256]
 690 str x[256],y[256],z[256]
 700 /*
 710 /* M M L   S E T
 720 /*
 730 a="v13l8o4":for i=3 to 8:m_trk(i,a):next
 740 a="@v127@71l1o3q8 c>baa-gf+f-g1&g1&g1":m_trk(1,a)
 750 a="@v127l8q8|:9r1:|r2ro2@76a16a16<@75e4":m_trk(2,a)
 760 a="@70c2&c>g4<cd2.g4f2&fc4fa-2.<d4e2&ec4ea2.b4<c2&c>a4<ce1
&e1&e1":m_trk(3,a)
 770 a="@70v12>g2&ge4gb2.<d4c2&c>a4<cd2.a4g2&ge4g<c2.e4f2&ff4fd
1&d1&d1":m_trk(4,a)
 780 a="@74v12l1ggfa-gaag1&g1&g1":m_trk(5,a)
 790 a="@74v12l1edcfef+fd&d&d":m_trk(6,a)
 800 a="@74v12l1c>ba<dcec>b&b&b":m_trk(7,a)
 810 a="@74v12l1r1r1r>br<crrrr":m_trk(8,a)
 820 /*
 830 a="<l8|:13cccc:|ccc>g&g1":m_trk(1,a)
 840 z="l4o2@76a<@75e>@76a8a8<@75e>":a="o2@76a<@75e>@76a8a8<@75
e8>@76a8&ar2<@75e"
 850 m_trk(2,z+z+z+z+z+z+a)
 860 a="e4.drgrf&f2.ef&ffe4derf&ffe4crdre4.drgrf&f2.ef&ffe4derc
&c1 ":m_trk(3,"o4@74v13l8"+a)
 870 c="@72v10l8o4|:13cc:|<v13crcc>v10|:12cc:|c>g&g2&ggv13l64b&
a&g&f&e&d&c>&b&a&g&f&e&d&c&b&ao4l8v10
 880 m_trk(4,c)
 890 b="c4.>br<drc&c2.cc&ccc4>b<crc&ccc4>arbr<c4.>br<drc&c2.cc&
ccc4>b<cr>a&a1
 900 m_trk(5,"o4@74v13l8"+b)
 910 m_trk(6,"@70v13o4l8"+a):m_trk(7,"@70v12o4l8"+b)
 920 a="@70v12o4l8>g4.grbra&a2.ga&aag4ggra&aag4frgrg4.grbra&a2.
ga&aag4ggrf&f1":m_trk(8,a)
 930 /*
 940 a="<cccccccc>eeeeeeeeffffffffggggg+g+g+g+aaaaaaaa<dddddddd
>gggggggg4rgrg4.<cccccccc>eeeeeeeeffffffffggggg+g+g+g+":m_trk(1
,a)
 950 a="|:7"+z+":|"+"o2@76a4<@75e8>@76a8r8a4.|:4"+z+":|":m_trk(
2,a)
 960 a="@70o5l8c4.>b4a4g&g2refgg4.f&f2e4.d4e4c&c2>b<cde4.ba4e4d
&d1r1<c4.>b4a4g&g2refgg4.f&f2e4.d4e4c&":m_trk(3,"@v127y50,20"+a)
:m_trk(5,"v12r16y52,28"+a)
 970 a="cccccccc>eeeeeeeeffffffffggggg+g+g+g+aaaaaaaa<dddddddd>
gggggggg<v13f4ecrd4.v10cccccccc>eeeeeeeeffffffffggggg+g+g+g+":m_
trk(4,a)
 980 a="l1@74o4v14edcd2d2edc@70v13<f4e8c8r8d4.>@74v14edcd2d2":m
_trk(6,a)
 990 a="l1@74o4v14c>bab2b2<c>aa@70v13<<c4c8>g8r8<c4.>@74v14c>ba
b2b2":m_trk(7,a)
1000 a="l1@74o3v14ggfg2g+2af+f@70v13<g4g8g8r8g4.>@74v14ggfg2g+2
":m_trk(8,a)
1010 /*
1020 a="aaaaaaaa<dddddddd>ggggggggggggggggf4ffffffffffffffffeeeee
eeeaaaaaaaa<ddddddddgggggga&a2&aeaebae>aaa<e>a":m_trk(1,a)
1030 y="o2@76a<@75e>@76a<@75e>":a=y+y+y+y+y+y+y+y:m_trk(2,z+z+z
+"o2@76a<@75e>@76a<@75e8>@76a8&"):m_trk(2,a)
1040 a="c2>b<cde4.dc4a4g&g1&g2redc&c2c4ed&d2d4ag&g1&g2r4rgg4fer
fre4.d4c4de&e1&e2r4re"
1050 m_trk(3,a):m_trk(5,a)
1060 a="aaaaaaaa<ddddddddgggggggggggggggf4ffffffffffffffffeeeeee
ee>aaaaaaaa<dddddddd>gggggg<e&e1&e1":m_trk(4,a)
1070 a="edc2&c8c4.>b2<d4.@73v13c8&cddc+cc2>a-4.<c+8&c+&c+":m_tr
k(6,a)
1080 a="<cc>a2&a8r4.g2b4.@73v13a8&abbaaa2f4.a8&a&a":m_trk(7,a)
1090 a="af+f2&f8r4.r2r4.@v127<a8&a2a2l2gag4.>b8r8<d4.c+c+8c+4.c
1cc4.eb4a4e4>b4<c+4>a4.<
1100 m_trk(8,a)
1110 /*
1120 d="<ddddddddeeeeeeee>eeeeeeeea4.ag4.gffffffffeeeeeeee<dddd
ddd>g4ggggggg4gggr4r<c&"
1130 m_trk(1,d)
1140 e=y+y+y+y+y+y+y:m_trk(2,e)
1150 f="o2@76a<@75e>@76a<@75e8>@76a<@75e8e8e8>@76r.a8&":m_trk(2
,f)
1160 g="f4efrarg&g2.rgg+4abrer<c&c2rcccc4>barbr<c2>g4g<cef4edrc
rd&d1&d2refg&
1170 m_trk(3,g):m_trk(5,g)
1180 h="ddddddddeeeeeeee>eeeeeeeeaaaaggggffffffffeeeeeeee<ddddd
dd>g4ggggggg4gggr4.<c&":m_trk(4,h)
1190 j="c>gg+aaga2..g8&g2..g8&g2r4.v14l8<e&":m_trk(6,j)
1200 k="aeeefef2..e8&e2..e8&e2r4.<v14l8c&":m_trk(7,k)
1210 l="l8f4.frf4.e4.drc4.e2e2a2a2a4.ara4.g4.grg4.a4.a2g2d4c8g4
&g4.<g>>@73v14r4.g&":m_trk(8,l)
1220 /*
1230 a="ccccccc>e4eeeeeeeggggggggaaaaaaa<d4ddddddc+4c+c+c+c+c+c
+c+cccccccc":m_trk(1,a)
1240 m_trk(1,">bbbbggg<c&"):m_trk(1,a+">bbbbeeea&a2..g+&g+2..g&
g2..f+&f+2..<f&")
1250 for i=1 to 2
1260 a="o2@76a8a8<@75e>@76a<@75e8>@76a4 a8<@75e>@76a<@75e":m_tr
k(2,a)
1270 a=y+"o2@76a<@75e>@76a<@75e8>@76a a8<@75e>@76a<@75e8>@76a a
8<@75e>@76a<@75e>"+y:m_trk(2,a)
1280 a="o2@76a<@75e>@76a<@75e8>@76a8&"
1290 b="@76a<@75e>@76a<@75e8>@76a8&|:4ar4<@75d>r8@76a8&:|
1300 if i=1 then m_trk(2,a) else m_trk(2,b)
1310 next
1320 a="g2refg&g2.r4r>b-<dargrf4fer4def&f2rdef&f2.r4r>a<cgrfre&
"
1330 b="eedr4efg&":c="eedr4edc&c2r>b<cd2.cde2.d+ea4.e4dcrf&
1340 m_trk(3,a+b+a+c):m_trk(5,a+b+a+c)
1350 a="ccccccc>e4eeeeeeeggggggggaaaaaaa<d4ddddddc+4c+c+c+c+c+c
+c+cccccccc
1360 m_trk(4,a+">bbbbggg<c&"+a+">bbbbeeea&a2..g+&g+2..g&g2..f+&
f+2..<f&")
1370 a="eedrerrd4dgrdr4.@74@v127d1c+2@73v14r4.f4ferfr4f4farfr4.
@74@v127f1
1380 m_trk(6,a+"g2b4.@73v14e&"+a+"d2b4.@73v14r4>a<ea&a2rcf<c&c2
>rcg<c&c2>rea<e2>v14c&")
1390 a="cc>br<cr4>b4b<dr>br4.@74@v127b-1a2a4.<v14@73d4dc+rdr4c4
cfrcr4.@74@v127c1
1400 m_trk(7,a+"d2g4.@73v14c&"+a+">b2g+4.v14a&a2..g+&g+2..g&g2.
.a&a2..@73v13a&")
1410 a="gggrgrrg4gbrgr4.@74@v127g1e2e4.@73v14a4aararra4a<c>rar4
.@74@v127a1
1420 m_trk(8,a+"b2<d4.>@73v14g&"+a+"g+2<e4.v14e&e2..f+&f+2..e&e
2..f&f2..a&")
1430 /*
1440 a="fferderf4ferd>gr<c4ccc|:12cccc:|ccc>g&g1<cccccccc>eeeee
eee":m_trk(1,a)
1450 a="l4a8a8<@75ee8>@76a8r8a8&a<@75ee8>@76a8r8a8&"+z+z+z+"@76
a<@75el16efdcl4>@76a":m_trk(2,a)
1460 a=z+z+"@76a<@75ee>@76a8a4.r2<@75e>":x="o2@76a<@75e>r8@76a8
<@75e>":m_trk(2,a+x+x)
1470 a="l8fferdcrf4ferderc&c1&c2.c8r8 @74v13o4rfe4derf4fe4crdre
4.drgrf&f2.ef4fe4derc&c4 v15@72y50,8rgbg<cdd+32&e8..&d+32&d16.>g
2d&ega&b<d&ega":m_trk(3,a)
1480 a="fferderf4ferd>gr<c4ccc|:5cccc:|ccv13<crcc>v10cc|:5cccc:
|cccd&d1cccccccc>eeeeeeee
1490 m_trk(4,a)
1500 a="l8fferdcrf4ferderc&c1&c2.&c8. v13@74o4rcc4>b<crc4cc4>ar
br<c4.>br<drc&c2.cc4cc4>b<cr>a&a4 v15@72y52,32rgbg<cdd+32&e8..&d
+32&d16.>g2d&ega&b<d&ega":m_trk(5,a)
1510 a="cc>gragr<c4c>graaro4@70v13e2drgrf&f2.ef4fe4derf4fe4crdr
e4.drgrf&f2.ef4fe4derc&c1v14o4@74l1ed
1520 m_trk(6,a)
1530 a="aaerfera4aerffro4@70v12c&c4.>br<drc&c2.cc4cc4>b<crc4cc4
>arbr<c4.>br<drc&c2.cc4cc4>b<cr>a&a1l1o4@74v14c>b":m_trk(7,a)
1540 a="aagragra4agraa ro3@70v12g&g4.grbra&a2.ga4ag4ggra4ag4grg
rg4.grbra&a2.ga4ag4ggrf&f1l1o3@74v14l1gg":m_trk(8,a)
1550 /*
1560 a="ffffffffggggg+g+g+g+aaaaaaaa<dddddddd>gggggggg4rgrg4.<
cccccccc>eeeeeeeeffffffffggggg+g+g+g+":m_trk(1,a):m_trk(2,x+x+x+
x+x+x+x+x+x+x)
1570 a="a+32&b8..<c>a4.rggfed&e&d>bg+ab4<c4b&ab<c4.>l16egrbagba
gf+fegf+fed+ded+ddcdccl64<g&g-&f&f-&e&d&dd-&d-&c&c&c-&c-&>&b&b&b
```

```
&&a&a&a-&a-& g4 &a&a+&b&<c&c+&d&d+&e& f&f+&g&g+>
 1580 m_trk(3,a):m_trk(5,a)
 1590 a="l8regeaebe<cdd+32&e..&d>g2&al16<cdc>agfec>afec>a4l8g<g4
>gg+<g+4>g+<"
 1600 m_trk(3,a):m_trk(5,a)
 1610 a="fffffffggggg+g+g+g+aaaaaaaa<dddddddd>gggggggg<v13f4ecr
d4.v10cccccccc>eeeeeeeefffffffggggg+g+g+g+":m_trk(4,a)
 1620 a="cd2d2edc<v14@70f4e8c8r8d4.>@74v15edcd2d2":m_trk(6,a)
 1630 a="ab2b2<c>aa<<v14@70c4c8c8r8c4.>v15@74c>bab2b2":m_trk(7,a
)
 1640 a="fg2g+2af+f<@70v14g4g8r4g4.>@74v15ggfg2g+2":m_trk(8,a)
 1650 /*
 1660 a="aaaaaaaa<dddddddd>ggggggggggggggf4fffffffffffffffeeeee
eeeaaaaaaaaa<ddddddddggggggga&a2&aeaebae>a4<aea":m_trk(1,a)
 1670 m_trk(2,x+x+x+"o2@76a<@75e>@76a<@75e8>@76a8&")
 1680 m_trk(2,y+y+y+y+y+y+y+"o2@76a<@75e>@76aa")
 1690 a="aega<cdcd&e&a&e&d&c2&>l16|:4g<cgc>:||:gb<g>b:|g4.":m_tr
k(3,a):m_trk(5,a)
 1700 a="o5l8@v127y50,20c4>a&gfedr<d4>b&agfer<e4&dc>b<c>b16&<c16
&>b16a16g+aea<c+>ea<cef4&edrcrg&g2g+>b<dfe4>b2<d4c+4>":m_trk(3,a
)
 1710 m_trk(3,"@l1aaaaa-a-a-a-gg ggg-g-g-g-ffff f-f-f-f-eeeedd d
dd-d-d-d-cccc c-c-c-c->bbbbaa a-a-a-a-ggggg-g- g-g-fffff-f-f-f-
eeeedddd-d- d-d-ccccc-c-c-c-> bbbbaa@70l8o4@v127re")
 1720 a="aaaaaaaa<dddddddd>gggggggggggggg<f4ffffffffffffffeeee
eeee>aaaaaaaa<dddddddd>ggggggga4.a4a4a4a1":m_trk(4,a)
 1730 a="o4l8v14y52,20a4f&edc>ar<b4g&fed>br<g4&feded16e16d16c16c
ec+eacea<c>a4&gfrer<d&d2d>fg+be4>b2<d4c+4r2l8o4@70v12rr16e":m_tr
k(5,a)
 1740 a="edc2&c8c4.>b2<d4.v11@73c8&cddc+cc2>a-4.<c+8&c+&c+":m_tr
k(6,a)
 1750 a="<c>aa2&a8r4.g2b4.v11@73a8&abbaaa2f4.a8&a&a":m_trk(7,a)
 1760 a="af+f2&f8r4.r2r4.<l2a8&aagag4.>b8r8<d4.c+c+8c+4.c1cc4.e2
b4a4e4>b4<c+4>a4.
 1770 m_trk(8,a)
 1780 /*
 1790 m_trk(1,">"+d):m_trk(2,e+f):m_trk(3,g):m_trk(4,"<"+h):m_tr
k(5,"y52,28"+g)
 1800 m_trk(6,"v13"+j):m_trk(7,"v13"+k):m_trk(8,"<@v127"+l)
 1810 /*
 1820 a="ccccccc>e4eeeeeeeggggggggaaaaaaa<d4ddddddc+4c+c+c+c+c+c
+c+cccccccc":m_trk(1,a)
 1830 m_trk(1,">bbbbggg<c&"):m_trk(1,a+">bbbbeeea&a2..g+&g+2..g&
g2..f+&f+2..<f&")
 1840 for i=1 to 2
 1850 a="o2@76a8a8<@75e>@76a<@75e8>@76a4 a8<@75e>@76a<@75e":m_tr
k(2,a)
 1860 a=y+"o2@76a<@75e>@76a<@75e8>@76a a8<@75e>@76a<@75e8>@76a a
8<@75e>@76a<@75e>"+y:m_trk(2,a)
 1870 a="o2@76a<@75e>@76a<@75e8>@76a8&"
 1880 b="@76a<@75e>@76a<@75e8>@76a8&|:4ar4<@75d>r8@76a8&:|
 1890 if i=1 then m_trk(2,a) else m_trk(2,b)
 1900 next
 1910 a="g2refg&g2.r4r>b-<dargrf4fer4def&f2rdef&f2.r4r>a<cgrfre&
 1920 b="eedr4efg&":c="eedr4edc&c2r>b<cd2.cde2.d+ea4.e4dcrf&
 1930 m_trk(3,a+b+a+c):m_trk(5,a+b+a+c)
 1940 a="ccccccc>e4eeeeeeeggggggggaaaaaaa<d4ddddddc+4c+c+c+c+c+c
+c+cccccccc
 1950 m_trk(4,a+">bbbbggg<c&"+a+">bbbbeeea&a2..g+&g+2..g&g2..f+&
f+2..<f&")
 1960 a="eedrerrd4dgrdr4.@74@v127d1c+2@73v14r4.f4ferfr4f4farfr4.
@74@v127f1
 1970 m_trk(6,a+"g2b4.@73v14e&"+a+"d2b4.@73v14r4>a<ea&a2rcf<c&c2
>rcg<c&c2>rea<e2>v14c&")
 1980 a="cc>br<cr4>b4b<dr>br4.@74@v127b-1a2a4.<v14@73d4dc+rdr4c4
cfrcr4.@74@v127c1
 1990 m_trk(7,a+"d2g4.@73v14c&"+a+">b2g+4.v14a&a2..g+&g+2..g&g2.
.a&a2..@73v13a&")
 2000 a="gggrgrrg4gbrgr4.@74@v127g1e2e4.@73v14a4aararra4a<c>rar4
.@74@v127a1
 2010 m_trk(8,a+"b2<d4.>@73v14g&"+a+"g+2<e4.v14e&e2..f+&f+2..e&e
2..f&f2..a&")
 2020 /*
 2030 a="fferderf4ferd>gr<c&":m_trk(1,a):a="o2@76a8a8<@75e8r8e8>
@76a8r8a8&a<@75ee8>@76a8r8a8&"
 2040 m_trk(2,a):a="fferdcrf4ferderc&":m_trk(3,a):m_trk(5,a)
 2050 a="fferderf4ferd>g<rc8&":m_trk(4,a)
 2060 a="cc>gragr<c4c>graar v12o4@70e&":m_trk(6,a)
 2070 a="aaerfera4aerffr v12o4@70c&":m_trk(7,a)
 2080 a="aagragra4agraar v12o3@70g&":m_trk(8,a)
 2090 /*
 2100 a="|:16cccc:|v15ccccv14cccc v13ccccv12cccc v11ccccv10cccc
v9ccccv8cccc v7ccccv6cccc v5ccccv4cccc v3ccccv2cccc v1ccccv0cccc
":m_trk(1,a)
 2110 a=z+z+z+"o2@76a<@75ee16f16e16>@76a16&a"+z+z+z+z:m_trk(2,a)
 2120 a="v15"+z+"v13"+z+"v11"+z+"v9o2@76a<@75ee16f16e16>@76a16&a
"+"v7"+z+"v5"+z+"v3"+z+"v1"+z:m_trk(2,a)
 2130 m_trk(3,"c2.&c2.&c8r8"):m_trk(5,"c2.&c2.&c8.")
 2140 a="ef&ffe4derf4fe4crdre4.drgrf&f2.ef4fe4derf4fe4crdr":m_tr
k(3,"o4v13@74"+a):m_trk(6,"e4.drgrf&f2."+a)
 2150 a="ef&ffe4derv11f4fe4crdv9re4.drgrv7f&f2.ev4f4fe4derv2f4fe
4crv1dr":m_trk(3,"e4.drgrf&f2."+a):m_trk(6,"e4.drgrf&f2."+a)
 2160 a="cc&ccc4>b<crc4cc4>arbr<c4.>b<rdrc&c2.cc4cc4>b<crc4cc4>a
rbr":m_trk(5,"o4v12@74"+a):m_trk(7,"c4.>br<drc&c2."+a)
 2170 a="o4cc&ccc4>b<cv11rc4cc4>arv9br<c4.>b<rdrv7c&c2.cv5c4cc4>
b<crv3c4cc4v2>arbv1r":m_trk(5,"c4.>br<drc&c2."+a):m_trk(7,"c4.>b
r<drc&c2."+a)
 2180 a="ccccccccccccccccccccccccc<v13crcc>v10cccccccccccccccc
cccccccccccccccc":m_trk(4,a)
 2190 a="v10ccccccccv8ccccccccv6ccccccccv4cccv6<crcc>v4ccv2cccccc
ccv1cccccccccv0cccccccccccccccc":m_trk(4,a)
 2200 a="g4.grbra&a2.ga4ag4ggra4ag4frgrg4.grbra&a2.ga4ag4ggra4ag
4frgr":m_trk(8,a)
 2210 a="v12g4.grbrv10a&a2.v8ga4ag4ggv6ra4ag4frgv4rg4.grbrv2a&a2
.ga4av1g4ggra4ag4frgr":m_trk(8,a)
 2220 /*
 2230 /*  P L A Y
 2240 /*
 2250 m_play()
```

日本音楽著作権協会(出)許諾第9070779-001

リスト2 ENDLESS RAIN1

```
10 '
20 '                 「 Endless Rain 」
30 '
40 '             from Album 'BLUE BLOOD'
50 '
60 '           music · lyric  :   YOSHIKI
70 '
80 ' Program by Y.Fushiki  1989/12/20 - 1990/5/27
90 '
100 VR=0
110 IF MEM$(&HAA09,2)=" ウ" THEN VR=400
120 'SOUND NUMBER 1 VOCAL
130 MEM$(&HB190+VR,36)=HEXCHR$("C3 00 71 32 71 22 0C 16 1B 00 1F
 1E 1E 1E 03 03 03 03 02 02 00 0A D4 A4 A7 A7 00 00 00 00 00 C8
80 00 03 80")
140 'SOUND NUMBER 2 GUITER
150 MEM$(&HB1B4+VR,36)=HEXCHR$("C2 00 51 11 52 12 09 12 0F 00 1F
 1E 1E 1E 03 03 03 03 02 02 00 0A D4 A4 A7 A7 00 00 00 00 00 C8
80 00 03 80")
160 'SOUND NUMBER 3 SNER DRUM
170 MEM$(&HB1D8+VR,36)=HEXCHR$("FC 00 0F 00 01 00 00 02 01 00 1F
 1F 1F 1F 00 8F 13 8B C0 80 C0 80 F5 F8 E6 C8 00 00 00 00 D4 C8
80 00 02 00")
180 'SOUND NUMBER 4 BASS DRUM
190 MEM$(&HB1FC+VR,36)=HEXCHR$("83 00 08 0F 00 00 01 00 07 00 1E
 1E 1E 1E 19 1B 11 0F 00 DF 00 00 FD AE F8 F8 00 00 00 00 DD C8
80 00 02 00")
200 'SOUND NUMBER 5 PIANO
210 MEM$(&HB220+VR,36)=HEXCHR$("FA 00 51 23 73 11 23 25 4D 00 1F
 1E 1D 1F 08 0F 12 8B 04 04 04 06 A5 56 56 A5 00 80 80 00 00 00
80 00 00 00")
220 'SOUND NUMBER 6 STRINGS 1
230 MEM$(&HB244+VR,36)=HEXCHR$("FC 00 01 02 00 01 1F 1B 39 00 01
 01 01 01 00 00 00 80 00 00 00 00 06 06 06 08 00 00 00 00 00 CC
80 00 02 00")
240 'SOUND NUMBER 7 HiHat OPEN 1
250 MEM$(&HB268+VR,36)=HEXCHR$("FC 03 33 3E 73 7F 00 00 07 00 1F
 9F 19 9F 0C 11 1C 1E 03 C6 02 C7 32 54 32 54 00 00 00 00 00 C8
80 17 03 00")
260 'SOUND NUMBER 8 HiHat OPEN 2
270 MEM$(&HB28C+VR,36)=HEXCHR$("FC 03 30 3E 71 7F 00 07 0E 00 1F
 9F 19 9F 0C 12 1C 1E 84 C6 C5 C7 32 84 32 74 00 00 00 00 00 C8
80 17 03 00")
280 'SOUND NUMBER 9 STRINGS 2
290 MEM$(&HB2B0+VR,36)=HEXCHR$("C4 00 64 24 68 68 2F 16 25 00 1F
 14 1F 14 00 00 00 00 00 00 00 00 06 00 06 00 00 00 00 00 C8
80 00 03 00")
300 'SOUND NUMBER 10 BASS
310 MEM$(&HB2D4+VR,36)=HEXCHR$("C2 00 70 52 20 30 1D 34 1B 00 19
 0B 19 12 06 08 09 87 01 01 01 01 F8 FA FA F7 00 00 00 00 00 96
80 00 02 00")
320 'SOUND NUMBER 11 TOM
330 MEM$(&HB2F8+VR,36)=HEXCHR$("FB 00 05 00 01 01 04 0F 1B 00 1F
 1F 1F 1F 1F 17 16 00 40 00 4C 0D 03 95 F8 08 00 00 00 00 00 00
80 00 00 00")
340 RUN"ENDLESS RAIN2.mml"
```

リスト3 ENDLESS RAIN2.mml

```
10 '   「 ENDLESS RAIN 」 Program No.2
20 'SAVE"ENDLESS RAIN2.mml"
30 INIT:CLS0:CLICKOFF:DEFINT A-Z:DEFSTR A,C,T:DIM A(40)
40 PLAY0:PLAY"T82
50 GOTO 120
60 LABEL "!"
70 READ I:IF I=-1 THEN 90
80 PLAY A(I);:GOTO 70
90 PLAY":";
100 RETURN
110 '==================================
120 '            VOCAL PART
130 '==================================
140 A(0)="I1 O3V9 Q8K0S2,2,0,11H3=1L8
150 A(1)="I6 O4V7 Q8 G1&G1&G1 V10
160 A(2)="G2G2 G2G2 G2G2 G2G2 C1 C2<B2> G2G2 G2G4.
```


▶LIVE in '90でX68000用のクラシック音楽をたくさん載せてほしい。ゲーム音楽はあんまり好きじゃないんです。「小フーガ ト短調」はよかった～。

山本 昭治(21)神奈川県

```
170 A(3)="_5G >EEEE16E16&E4<GG >DDDD16D16&D4R4 RCCC16C16&C<BAB16
>C16& C4D4C4<B4>
180 C="EEEE16E16&E4R16<G16G16G16 >DDDD16D16&D4R4
190 A(4)=C+"R4<AA>AAAA16G16& G4RDE4D4
200 A(5)="C4RED.C16&C<B >C4RDE4D4 C4RED4C<B16>C16& C2R2 R2R4.
210 A(6)="<G16G16>EEEE16E16&E4R<G16G16 >DDDD16D16&D4R4 RCCC16C16
&C<BAB16>C16& C4D4C4<B4>
220 A(7)=C+"R4.<A16A16>AAA16AA16 G4RDE4D4
230 A(8)="C4RD16D16EDC<B16>C16& C4RDE4D4 C4RED4C<B16>C16& C4R4R2
240 A(9)="E2.B4> C2R4D16C<B16> C2R4D16C<B16> C4RDC4<B4
250 A(10)="E4E4R4B4> C2R4D16C<B16> C2R4D16C<B16> C2<B2 I10O3V11K
2
260 A(11)="C4S3,3,0,29H3=1G4=0>C4.D16&C16 F16&G4.&G16&G&F<S4,1,0
,6=1B4=0 A2.&AG16E16 F>FL16CFRF32&G32RDR<B8GL8=1D=0
270 A(12)="C4S3,3,0,29=1G4=0>C4.G16&A16 S4,1,0,6=1G2=0<C4.<B> G1
6&A16&A4>D16E16F.E.D16C16 <G>L16C&DGDG&A
280 A(13)=">E4D4
290 A(14)="C4RED4C<B16>C16& C4RDE4D4 C4RED4C<B16>C16& C4R4R2
300 A(15)=">>C2R<A-B->C< B-1 >C2RCDE- D4R4D4D4
310 A(16)="E-1 I2 V9 =0O3E32&F16.Q0{E-FE-}8'Q8DS3,2,0,6H3=1C=0L1
6<B->CDE32&F32E-DC<B- B-&A-8.>=1C8=0<B-8A-GFE-FGA-B-L8> S3,2,0,4
=1C=0<B->=1E-=0D=1G=0F>=1C=0<B-
320 A(17)="S3,2,0,6=1>D16&=0E-.RF16&E-16D.<G16>DE-16F16 E-16&D16
CRD16&C16<B-4>F32&G16.F16E-16 <A-4.=1G16&=0A-16G4L16FGA-B- >C<A-
B->CD<B->CDE-CDE-=1A32&=0B-32GA-B-L8
330 A(18)="<R4G4FGA-B- G4E-<B-B>DFA- G16F16E-4.DE-FA- G4>B-Q0{A-
B-A-}8'{GA-G}8'{FGF}8'{E-FE-}8'{DE-D}8'Q8
340 A(19)="C4F16G.FGA-B-16&A-16 G4FG<L16B>DFA-GFE-F E-FGA-B-GA-B
->C<A-B->CD<B->CDL8
350 A(20)="C2.Q0S4,1,0,5C32=1D&D32=0C16Q8'< B2.&B."+A(0)+"G16> E
4E.F16&FER<G16G16> DDDD16D16&D4R4
360 A(21)="RCCC16C16&C<BAB> C4D4C4<B4> EEEE16E16&E4R16<G16G16G16
> DDDD16D16&D4R4
370 A(22)="R4R<A16A16>A4AA G4RDE4D4 C4RD16E16&EDC<B16>C16&
380 A(23)="C4RDE4D4 C4RED4C<B16>C16& C4R4R2 R1
390 C="E2.B4> C2R4D16C<B16>
400 A(24)=">"+C+"C2R4D16C<B16> C4C4<B4R4
410 A(25)=C+"C2R4D16C<B16> C4.C<B4R4
420 A(26)=C+"\ C2R4D16C<B16> C4RDC4<B4"      'Fade Out
430 A(27)="E4E4R4B4> C2R.DC<B16> C2R.DC<B16> C2<B4R4
440 '
450 "!
460 DATA 0,1,2,0,3,4,5,6,7,8,9,10,11,12,0,13,14,9,10,0,15,16,17,
18,19,20
470 DATA 21,22,23,9,10,0,24,25,26,27,-1
480 '===================================
490 '           VOCAL PART ECHO
500 '===================================
510 A(0)="I1 O3V5 Q8K2S2,2,0,18H3=1L8
520 A(1)="I6 O4V7 Q8 D1&D1&D1 V9
530 A(2)="E2D2 C2D2 C2<B2 A2B2 F1 F2E2 >C2C2 C2C4. R
540 A(6)="CCCC16C16&C4RR<G16G16 >DDDD16D16&D4R4  EEE16E16&EDCD16
E16& E4F4E4D4R
550 A(8)="C4RD16D16EEF16G16& G4RGG+4G+4 A4RAG4GG16G16& G4R4R2R
560 A(10)="E4E4R4B4> C2R4D16C<B16> C2R4D16C<B16> C2<B2 I10O3V7 K
2
570 A(16)="E-2.&E I1 V8 =0O3E32&F16.Q0{E-FE-}8'Q8DS3,2,0,6H3=1C=
0L16<B->CDE32&F32E-DC<B- B-&A-8.>=1C8=0<B-8A-GFE-FGA-B-L8 S3,2,0
,4=1G=0F>=1C=0<B->=1C=0<B->=1B-=0A-
580 A(17)="S3,2,0,6>G4RA-16&G16F.<B-16>FG16A-16 G16&F16E-RF16&D1
6D4A-32&B-16.A-16G16 C4.<=1B16&>=0C16<B-4L16A-B->CD E-CDE-FDE-FG
E-FGB-GA-B-L8
590 A(18)="<R4E-4DE-FG E-4<B-GGB>DF E-16D16C4.<B->CDF E-4>GQ0{FG
F}8'{E-FE-}8'{DE-D}8'{CDC}8'{<B>C<B>}8'Q8
600 A(19)="<A-4>D16&E-.DE-FG16&F16 E-4DE-<L16GB>DFE-DCD CDE-FGE-
FGA-FGA-B-GA-B-L8
610 A(20)="G2.Q0S4,4,0,5A-32=1B-&B-32=0A-16Q8' G1"+A(0)+">C4C.D1
6&DCRR<G16G16> DDDD16D16&D4R4
620 A(22)="R4R<A16A16>A4AA G4RDE4D4 C4RD16E16&EDF16G16&
630 A(23)="G4RGG+4G+4 A4RAG4GG16G16& G4R4R2 R1 R
640 C="C2.G4 A2R4B16AG16 A2R4B16AG16
650 A(24)=C+"A4A4G4R4
660 A(25)=C+"A4.AG4R4
670 A(26)=C+"A4RBA4G4
680 A(27)="C4C4R4G4 A2R.BAG16 A2R.BAG16 A2G4R4
690 A(28)="E4E4R4B4> C2R4D16C<B16> C2R4D16C<B16> C2<B4.
700 '
710 "!
720 DATA 0,1,2,0,3,4,5,6,7,8,9,10,11,12,0,13,14,9,10,0,15,16,17,
18,19,20
730 DATA 21,22,23,9,28,24,25,26,27,-1
740 '===================================
750 '           Piano1
760 '===================================
770 A(0)="I5 O4V11Q8L8K0 R1R1R1
780 A(1)=">E<G>E<G>D<G>D<G >E<A>E<A>D<G>D<G
790 A(2)="AC>C<F>F<A>AC <AC>C<F>D4<G32>D32E16F
800 A(3)="F<G>F<G>E<G>E<G> F<G>F<G>E<G>E<G
810 C="E<G>CD>C<EGC D<GB>DBDDF
820 A(4)=C+"E<A>CE>C<EAC F<A>F<A>D<G>D<G>
830 A(5)=C+"E<A>CEACAC D<G>D<G>E<G>D<G>
840 A(6)="F<A>F<A>G<B>G<B> GCGCG+<B>G+<B>
850 A(7)="ACACG<B>G<B> F<G>F<G>E<G>E<G> F<G>F<G>E<G>E<G>
860 A(8)=A(4)
870 A(9)="E<G>CE>C<EGC D<GB>DBDDF E<A>CEACAC D<G>D<G>E<G>D<G>
880 A(10)=A(6)+"ACACG<B>G<B> F<G>F<G>GC16G16GC
890 C="E<G>E<G>D<G>D<G> E<A>E<A>E<A>E<A> F<A>F<A>GCGC
900 A(11)=C+"ACACD<G>D<G>
910 A(12)=C+"ADADF<B>F<B>
920 A(13)="E<G>CL16E8.CEGCEG>CL8 D<DGB>R16D16<D>GF E<EABL16>C<BA
B>C<BL8A FC.>E.D.<B>C<B16
930 A(14)="L16>C2&C<EG>CE<G>CEL8 D2R16B>R16CD L16C4&CEC<A>{C<AEA
EC}4{EC<A>C<AE}4L8 D4&D16>D16<G>E<G>D<G
940 A(15)=">F4<C16F16A>F.GDE16& E<G>E<G>E<G+>D<G+ >C4<C16F16A16F
16>F.GB. >C4<F<G>E<G>CG<
950 A(16)=A(11)
960 A(17)="E<G>E<G>D<G>D<G> E<A>E.>C16C<CAC ACACGCGC ADADG<B>G<B
>
970 A(18)=STRING$(2,"A-4A-4>C4E-4< B-4B-.>D16D.F16F4<")
980 C="G<B->G<B->F<B->F<B-> E-<G>E-<G>D<F>D<F> E-<A->E-<A->E-<G>
E-<G>
990 A(19)=C+"F<A->F<A->F<B->F<B->
1000 A(20)=C+"F<A->F<A->E-<A->F<B->
1010 A(21)=STRING$(3,"E-<A->E-<A->F<B->F<B-> G<B->G<B->G<B>G<B>"
)
1020 A(22)=">C<E->C<E->D<F>D<F
1030 A(23)="R<G>CDG>CDG D<GB>DB2< E<G>CE>C<G>C<E D<GB>DBDDF
1040 A(24)="E<G>CE>C<EAC F<A>F<A>D<G>D<G> E<G>CE>E<G>C<E G<B>DG>
D<GDA-
1050 A(25)="ACEA>C<F>C<F> C<D>C<DBDBD ACACBDBD
1060 A(26)=">C<E>C<EBEBE ACACBDBD G4G4G4&G16>C. C4C4C.C.C<
1070 C="E<G>E<G>D<G>D<G> E<A>E<A>E<A>E<A>
1080 A(27)=C+"F<A>F<A>GCGC ADADD<G>D<G>
1090 A(28)=C+"F<G>F<G>GCGC ADADF<B>F<B>
1100 A(29)="E<G>E<G>D<G>D<G> E<A>EC16>C16C4<A4 ACACGCGC ADADG<B>
G<B>
1110 A(30)=C+"ACACGCGC ADADG<B>G<B>
1120 '
1130 "!
1140 DATA 0,1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
,21,22,23
1150 DATA 24,25,26,27,28,29,30,30,30,-1
1160 '===================================
1170 '  String & PIANO ECHO & etc
1180 '===================================
1190 A(0)="I6 O3V9 Q8K0L8 C1&C1&C1& V11C1&C1&C1&C1 <F1G1> C1&C1
I5 O4V7 Q8L8K2 R16
1200 A(10)=A(10)+"16
1210 A(11)="S2,2,0,10H3=1I9O0V8 G1 A1 A2>C2 F2FED4
1220 A(12)="E1 E1 F2G2 A2B2=0
1230 A(13)="I5O4V7 R16E<G>CL16E8.CEGCEG>L8 D.<DGB16R32B16.D>DD C
R16<EABL16>C<BAB>C<BL8A FC>C.<B.GGG16
1240 A(14)="L16G2&GR16EG>CE<G>CL8 <B2>R32G&G32R32G&G32B L16A4&AR
16>EC<A>{C<AEAEC}4{EC<A>C<AE}4L8 D4&D16>D16<G>E<G>D<G16
1250 A(15)=">C4R16<C16F16A16>C.D<B>C16& CR16<G>E<G>E<G+>D<G+16 >
C4R16<C16F16A16>C.DG. G4R16F<G>E<G>CG16
1260 A(16)="S2,2,0,10H3=1I9O1V8 E2D2 C2D2 F2G2 A2D2
1270 A(17)="E2D2 C2E2 A2G2 A2B2
1280 A(18)=">C2&C<A-B->C D2.<B-4> C2&CC16E-16GG F2F4=0I2 O3V10CD
1290 A(19)="E-4A16&B-.B-16&A-16GA-Q0{GA-G}8'Q8 S2,2,0,10=1I9O2V8
 C2<B-2 A-2B-2> C2D2
1300 A(20)="E-2F2 G2D2 C2E-2 F2<B-2
1310 A(21)="I2 O4V9 =0A-2B-2 I9O2V8 =1B-2B2> C2D2 E-2D2
1320 A(22)="C2D2< B-2B2 A-2B-2
1330 A(23)="I2 O5V7 =0R16C2.Q0S4,1,0,5R16C32=1D&D32=0C16Q8'< B2.
&B I5O4V7 R16E<G>CE>C<G>C<E D<GB>DBDDF
1340 A(26)=">C<E>C<EBEBE ACACBDBD16 >C4C4C4&C16E. F4F4E.E.E
1350 A(27)="S2,2,0,10H3=1I9O1V8 E2D2 C2E2 F2G2 A2G4F4
1360 A(28)="E2D2 A1 A2G2 A2B2
1370 A(29)="E2D2 A1 F2G2 A2B>CD<B
1380 A(30)=">C2<B2 >C1 F2G2 A2B2
1390 A(31)="E1 E1 F2G2 F2G4D4
1400 A(32)="C2<B2 >C1 F2G2 A2B2
1410 '
1420 "!
1430 DATA 0,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,2
3
1440 DATA 24,25,26,27,28,29,30,31,32,-1
1450 '===================================
1460 '           Piano2
1470 '===================================
1480 A(0)="I5 O4V11Q8L8K0 R1R1R1
1490 A(1)=">CRCR<BRBR >CRCR<BRBR
1500 A(2)="FRAR>CRFR <FRARF4R64B32R64R16R
1510 A(3)=">CRCRCRCR CRCRCRCR<
1520 C="CRRRGRER <BRRR>GR<BG+
1530 A(4)=C+">CRRRARER CRDR<BRBR>
1540 A(5)=C+">CRRRFRFR CRCR<BRBR>
1550 A(6)="CRCRDRDR ERERERER
1560 A(7)="FRFRDRDR CRCRCRCR CRCRCRCR
1570 C="CRRRGRER <BRRR>GR<BG+>
1580 A(8)=C+"CRRRARER CRDR<BRBR>
1590 A(9)=C+"CRRRFRFR CRCR<BRBR>
1600 A(10)="CRCRDRDR ERERERER FRFRDRDR CRCRER16E16ER
1610 C="CRCR<BRBR> CRCRCRCR CRCRERER
1620 A(11)=C+"FRFRCR<BR>
1630 A(12)=C+"FRFRDRDR
1640 A(13)="CRRD.R16RRR GRRRGRBB ARR2. R<A.>F.G.DED16
1650 A(14)="E2&E16R16RRR G2>D.E.G E4&L16E>C<AE{AECEC<A}4>{C<AEAE
C}4 <B4&B>BL8RBRBR
1660 A(15)="A4R4A.BGG16& GR>CR<BRBR F4R4A.B>D. F4CRCR4.<
1670 A(16)=A(11)
1680 A(17)="CRCR<BRBR> CRC.E16ERER FRFRERER FRFRDRDR
1690 A(18)=STRING$(2,"C4C4E-4A-4 D4D.F16F.B-16B-4")
1700 C="E-RE-RDRDR CRCR<B-RB-R> CRCR<B-RB-R>
1710 A(19)=C+"CRCRDRDR
1720 A(20)=C+"CRCRCRDR
1730 A(21)=STRING$(3,"CRCRDRDR E-RE-RDRDR")
1740 A(22)="A-RA-RB-RB-R
1750 A(23)="I7O5V16D2R4.V15{BBB}8 B2I5O5V11G2< CR2RGR <BRRR>GR<B
G+>
1760 A(24)="CRRRARER CRDRCR<BR> CRRR>CR<GR DRRRBRR<B>
1770 A(25)="ERRRARAR GRGRGRGR FRFRGRGR
1780 A(26)="GRGRG+RG+R FRFRGRGR F4F4E4&E16G. G4G4G.G.G
1790 A(27)="CRCR<BRBR >CRCRCRCR CRCRERER FRFRCR<BR>
1800 A(28)="CRCR<BRBR >CRCRCRCR CRCRERER FRFRDRDR
1810 A(29)="CRCR<BRBR >CRRR16E16E4C4 FRFRERER FRFRDRDR
1820 A(30)="CRCR<BRBR >CRCRCRCR FRFRERER FRFRDRDR
1830 '
1840 "!
1850 DATA 0,1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
,21,22,23
```

▶僕ははっきり言ってバカです。パソコンに関してだけど……。それと一言，「西川善司バンザイ！」。
備後　秀明(16)群馬県

```
1860 DATA 24,25,26,27,28,29,30,30,30,-1
1870 '==================================
1880 '            Piano3
1890 '==================================
1900 A(0)="I5 O3V12Q8L8K0 R1R1R1
1910 A(1)="C1 C2.&C<G>
1920 A(2)="F4.CF2 <G2G4BG
1930 A(3)=">C4.<G>C4.<G >C4.<G>C4.<G
1940 C=">C2.&CC C2.&C<B
1950 A(4)=C+"A2.&AA F4.FG4.G
1960 A(5)=C+"A2F4.F G4.GG4.G
1970 A(6)="F4.FG4.G >C4.<G>E4.<B
1980 A(7)=">F4.CG4.D C4.<G>C4.<G> C4.<G>C4.<G>
1990 C="C2.&CC C2.&C<B
2000 A(8)=C+"A2.&AA F4.FG4.G>
2010 A(9)=C+"A2F4.F G4.GG4.G>
2020 A(10)="<F4.FG4.G> C4.DE4.<B> F4.CG4.D C4.<G>GC16G16&GC
2030 C="C4.C<B4.B A4.AG4.G
2040 A(11)=C+">F4.FE4.E D4.D<G4G4>
2050 A(12)=C+" F4.FE4.E D4.DG4G4>
2060 A(13)="C1 C2.<GG+ A1 F2G2>
2070 A(14)="C1 C2.&C<B A1 G2>G4G4
2080 A(15)="<F2G2 >C2<E2 F2G2 >C1
2090 A(16)=A(11)
2100 A(17)="C4.G16>C16<<BB4G16G+16 A4.>AGG4D16E16 F4.C16F16EE4C1
6C+16 D4.F16F+16G4<G4
2110 A(18)=STRING$(2,STRING$(4,"A->E-<")+STRING$(4,"A->D<"))+">"
2120 C="E-4.E-D4.D C4.C<B-4.B- A-4.A-G4.G
2130 A(19)=C+"F4.FB-4.B->
2140 A(20)=C+"F4.FB-4B-4>
2150 A(21)=STRING$(3,"<A-4.A-B-4.B-> E-4.E-<G4.G>")
2160 A(22)="<A-4.A-B-4.B->
2170 A(23)="<G1& G1> C2.&CC C2.&C<B>
2180 A(24)="<A2.&AA F4.FG4.G >C2.&CC C2.&C<B>
2190 A(25)="<A4.AF4.F G4.GG4.G F4.FG4.G>
2200 A(26)="C4.CE4.E <F4.FG4.G >C4G>C<<CE16G>C. C4.<G>CC4.
2210 A(27)="C4.C<B4.B A4.AG4.G >F4.FE4.E D4.D<G4G4>
2220 A(28)="C4.C<B4.B A4.AG4.G F4.FE4.E D4.DG4G4>
2230 A(29)="C4.C<B4.B A4.AG4G4 F4.FE4.E D4.DG4G4>
2240 '
2250 "!
2260 DATA 0,1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
,21,22,23
2270 DATA 24,25,26,27,28,29,28,28,28,-1
2280 '==================================
2290 '            BASS
2300 '==================================
2310 A(0)="I10O3V11Q8L8P3K0 R1R1R1
2320 A(1)=">>S4,3,0,6H3=1C4.=0<G16&A16G4.C16&D16 C2F16&G.C16<G16
>C16&D16
2330 A(2)="C16&D16C&C2.& C2.&CF16&G16
```

```
2340 A(3)="F2&F<A>CG16&A16 G2<S3,3,0,29H5=1G2=0 >C1& C1
2350 A(4)="R1R1R1R1
2360 A(5)="S4,1,0,6R4.>=1C&C16&=0<G.&=3G4=0
2370 A(6)="<C1 C2.&CG+ A1 F4.F16F+16G4<GA16G16
2380 A(7)=">C1 C2.&C<B A2>F4.F G2G.>D16&D4
2390 A(8)="<<F4.F+GG4B >C4.DE4.E F4.F16F+16G2 C4.S3,3,0,40=3G=0>
C2
2400 A(9)="<C2<BB4B A4.AGG4>E F2EE4E16C16 D4.AG4<GB>
2410 A(10)="C4.G16C16<BB4G16G+16 A4.>AGG4G16E16 F4.C16F16EE4E16C
16 D4.F16F+16GG16D16<G4>
2420 A(11)="C4H3=1G4=0>C4.D16&C16 F16&G4.&G16&G&F<S4,1,0,6=1B4=0
 A2.&AG16E16 F>FL16CFRF32&G32RDR<B8GL8=1D=0
2430 A(12)="C4S3,3,0,29=1G4=0>C4.G16&A16 S4,1,0,6=1G2=0<C4.<B> G
16&A16&A4>D16E16F.E.D16C16 <G>L16C&DGDG&ABABG8FEDL8<
2440 A(13)="F4.F16F+16GG4<B16B16> C4.D16D+16E4<E4 F4.F16F+16G4.G
> C4.S3,3,0,29=1G=0>C2<
2450 A(14)="C4.<BB4.G A4.>AGG4D16E16 F4.C16F16EE4E16C16 D4.AG4<G
A16B16>
2460 A(15)="C4.G16>C16<<BB4G16G+16 A4.>AGG4D16E16 F4.C16F16EE4C1
6C+16 D4.F16F+16G4<G4>
2470 A(16)="A-A-A-A-A-A-A-A-
2480 A(16)="<"+A(16)+">"+A(16)+"<"+A(16)+">A-A-A-A-<A-4>A-4
2490 C="E-4.B-16E-16DD4<B-16B16> C4.G16>C16<B-B-4B-16G16
2500 A(17)=C+"A-4.E-16A-16GG4G16E-16 F4.>C<B-4<B->F16<B-16>
2510 A(18)=C+"A-4.>E-16<A-16GG4E-16E16 F4.A-16A16B-4<B-16BB-16>
2520 A(19)="<A-4.A-16A16B-B-4>C16D16 E-4.<F16F+16GG>G<G A-4.A-16
A16B-B-4>C16D16 E-4.F16F+16G&>B16>C16D<S4,1,0,6=1G<=0
2530 A(20)="A-4.A-16A16B-B-4C16D16 E-4.F16F+16G4<G4 A-4&L16A->E-
A-AB-FB-8B-8FF+L8
2540 A(21)="G2.&G16>S3,3,0,40=1D.=0 G1< R1R1 R1R1R1R1
2550 A(22)="S4,3,0,6H4=1>A2F2 G4&G.=0D16G16D16<G16>D16<GG <F4.F1
6F+16GG4A16B16>
2560 A(23)="C4.D16D+16E4<E4> F4.F16F+16G4=2G=0<G> C4&C16S3,1,0,8
=2G.=0>C4&C16&G.& >C4&C16<S4,1,0,1=1G.=0C2<
2570 A(24)="C4.G16>C16<<BB4G16G+16 A4.>AGG4D16E16 F4.>C16F16<EE4
C16C+16 D4.F16F+16G4<GB16G16>
2580 A(25)="L16C4.GC<B8B4GG+ A4.>EAGAG4DE F4.>C<FE8E4CC+ D8>CC+D
8<FF+G8<G8G8AB>
2590 C="C4.GC<B8B4GG+ A4.>EAG8G4DE
2600 A(26)=C+"F4.CFE8E4EC D8CC+D8.>CEDR<B8G<AB>
2610 A(27)=C+"F4.>C<AE8E4CC+ D8>CC+D<ADAG8D8<GABG>
2620 A(28)="C4.G>C<<B8B4GG+ A4.>A8G8G4DE F4.>C<FE8E4. D8A>D8<DA8
GD8.<G8AB>
2630 A(29)="C4.GC<B8B4GG+ A8AA8>A8.GAG4DE F8F4>CF{EC<G}4{ECC+}4
D4.FF+=2G4=0<G4
2640 '
2650 "!
2660 DATA 0,1,2,3,4,4,4,5,6,7,8,9,10,11,12,13,14,15
2670 DATA 16,17,18,19,20,21,22,23,24,25,26,27,28,29,-1
2680 '
2690 CHAIN"ENDLESS RAIN3.mml"
```

リスト4　ENDLESS RAIN3.mml

```
10 '    「 ENDLESS RAIN 」 Program No.3
20 'SAVE"ENDLESS RAIN3.mml"
30 RESTORE
40 GOTO 100
50 LABEL "!"
60 READ I:IF I=-1 THEN 80
70 PLAY A(I);:GOTO 60
80 PLAY":";
90 RETURN
100 'TOM ﾊﾟﾀｰﾝ
110 T1="I11V11L64O3D&C&<B&AV12O0L8
120 T2="I11L64O2G&F&E&DO0L8
130 T3="I11V13L64O2C&<B&A&GV12O0L8
140 T7="I11V11L64O3D&CV12O0L8
150 '==========================
160 '  BASS ECHO & Drums
170 '==========================
180 C="I4B4I3F
190 A(0)="I10O3V11Q8L8P3K3 R1R1R1
200 A(5)="R4.I11V13O3D64D32.&C32&<B32R16"+T1+T2+"R16"
210 A(6)=T2+T2+T3+T3
220 A(7)="O0V12S4,1,0,60I4BBR4I3F4R4
230 A(8)="I4BB16B16I3F.I4B16I3F16"+T1+"R16"+T2+"R16"+T2+T3+T3
240 A(9)=C+"4I4BBI3F4
250 A(10)=C+"4I4B16B16R16B32B32I3F16F16R16F16
260 A(11)=C+"4I4B16B16R16B16I3F4
270 A(12)=C+".F16I4B16I3F16R16"+T1+"R16"+T2+T2+T2
280 A(13)=C+"4I4B16B16R16B16I3F16I4B16R16B16
290 A(14)="I4BBI3F16F16R16I4B32B32I3F16F16R16"+T1+"R16"+T2+T3+T3
300 A(15)="I4BBI3F4I4B16B16R16B32B32I3F16F16R16F16
310 A(16)="L16I4B8B8I3F8.F8I4BI3FI4B32B32I3FFRFL8
320 A(17)="L16I4B8B8I3FI4BRBRB32B32I3FFI4BI3F8I4BL8
330 A(18)="I4BB16B16I3F.F16I4B16I3F16I4B16B16I3F.F16
340 A(19)="I4B16I3F16I4B16B16I3F.F16{RFFFFF}4I11V13O3=3{DD<GGCC}
4O0=0V12
350 A(20)=C+"4I4B16B16R16B16I3FI4B16B16
360 A(21)=C+".F16I4B16I3F16"+T1+"R16"+T1+T1+T2+T2
370 A(22)="I4B4R2R8I8O4V16{GGG}8 G2V12R2 O0R1 R1
380 A(23)="R1R1R1R1 I7O5V3 {AAA}8V6 {AAAAAA}4V8 {AAA}8V11{AAA}8V
12{AAAA}8V14{AAAAA}8V15G16G16
390 A(24)="I7O5V15DI4O0V12B16B16I3F.I4B16I3F16"+T1+"R16"+T2+"R16
"+T3+T3+T3
400 A(25)=C+"4I4B16B16R16B16I3F.F16
410 A(26)="I4B.B16I3F.F16I4B16I3F16R16"+T7+T7+T1+T2+T3+T3
420 A(27)=C+"4I4BB16B16I3FI4B
430 A(28)="I4B.B16I3F4F16F16R16F16"+T1+T2+T1+T1
440 A(29)=C+"4I4BB16B16I3F4
450 A(30)="I4BB16B16I3F.F16I4B16I3FI4B16I3F16F16R16F16
460 A(32)="@L8I4B&BBI3FFFI4BI3FFFI11O3=3D<G=0I3O0FI11O2=3GC=0I3O
0FFFFFI11O3=3DD<GG=0O0L8
470 A(33)=C+".F16I4B16I3FI4B32B32I3F16F16R16F16
480 A(34)="I4B.B16I3F.F16I4BBI3F.F16
490 A(35)="L32I4B16I3FFFFI4B16I11O2=3AAAA=0I4O0B16I11O2=3DDDD=0I
4O0B16I11O1=3AAAA=0@L8I4O0BI11O1=3AAAAA=0O0L8
500 A(36)="L16I4B8BBI3FFRFI4BI3F8I4B32B32I3FFRFL8
510 A(37)="I4BB16B16I3F.F16I4BBI3F.F16
520 A(38)="L16I4BBRBI3FFRFI4BI3F.R4I11O3=3DD=0
530 '
540 "!
550 DATA 0,1,2,3,4,4,4,5,6,7,7,7,7,7,7,8,9,9,9,10,9,9,9,11,9,9
,9,12
560 DATA 9,9,9,11,9,9,9,13,9,9,9,14,9,9,9,13,9,9,9,15,16,17,18,1
9
570 DATA 9,9,9,20,9,9,9,13,9,9,9,11,9,9,21,22,23,24,9,9,9,25,26
580 DATA 9,9,9,13,9,27,9,28,9,29,9,30,9,9,16,32,9,33,34,35,9,36,
37,38,-1
590 '==========================
600 '   PSG Hi Hat
610 '==========================
620 A(0)="V16_3O8L8y7,49y6,8S4,1,30,0^1=3
630 A(1)="R1R1R1R1
640 A(2)="S4,3,20,0y6,4¯C4_S4,1,30,0y6,6AAAAAA
650 A(3)="y6,6AAAAAAAA AAAAAAAA
660 A(4)="y6,6AAAAAAA16A16=0y7,48Cy7,49=3
670 A(5)="y6,6AAAAAAAA
680 A(6)="y6,6AAAAAAA4
690 A(7)="y6,6AAAAA4R4
700 A(8)="R4S4,3,20,0y6,4¯C4_S4,1,30,0R2
710 A(9)="S4,3,20,0y6,4¯C4C4_S4,1,30,0R2
720 A(10)="S4,3,20,0y6,4¯C8.C4&C16C4R16C8.C4_S4,1,30,0
730 A(11)="S4,3,20,0y6,4¯C4C4_S4,1,30,0y6,6A16A16S4,3,20,0y6,4¯C
4_S4,1,30,0
740 A(12)="y6,6AAAAA=0y7,48C16C32C32C16C16y7,49=3R8
750 A(13)="y6,6AAAAS4,3,20,0y6,4¯C4_S4,1,30,0y6,6A4
760 A(14)="S4,3,20,0y6,4¯C4_S4,1,30,0y6,6AAS4,3,20,0y6,4¯C4_S4,1
,30,0R4
770 A(15)="S4,3,20,0y6,4¯C2_S4,1,30,0R2
780 '
790 "!
800 DATA 1,1,1,1,1,1,0,2,3,4,5,3,13,2,3,5,2,3,5,2,3,5
810 DATA 2,3,4,2,3,5,2,3,6,2,3,5,2,3,5,2,2,2,7
820 DATA 2,3,5,2,3,5,2,3,5,2,5,14,15,1,1,8,2,3,5,6
830 DATA 2,3,5,2,3,7,2,3,5,2,3,9,2,5,2,10,2,5,11,12,-1
840 '
850 PLAY ""            :'MUSIC START
```


▶ポピュラスやると性格が悪くなるので，みなさんいっしょにやりましょう。

船戸　隆(16)岐阜県

リスト5 Running up!

```
10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
20 /*                                                  /*
30 /*             R u n n i n g   u p !                /*
40 /*                                                  /*
50 /*              (C) NAMCO / MEGATEN                 /*
60 /*                                                  /*
70 /*           Arranged by Z.NISHIKAWA                /*
80 /*                                                  /*
90 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
100 m_init()
110 dim char v(4,10)
120 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN      S.E.1
130 v={59, 15,  2,  1,200,127,  0,  0,  0,  3,  0,
140 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
150     31,  8,  1,  8,  7, 20,  2,  1,  5,  3,  0,
160     31,  8,  8,  7,  5, 24,  1,  2,  1,  1,  0,
170     31,  3,  7,  8,  1, 21,  1,  1,  3,  0,  0,
180     31,  0,  0,  9,  0,  0,  2,  8,  5,  2,  0}
190 m_vset(1,v)
200 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN      S.E.2
210 v={56, 15,  2,  1,100,127,  0,  4,  0,  3,  0,
220 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
230     20, 10,  0,  5,  0, 44,  0,  0,  0,  0,  0,
240     20, 16,  0, 10, 15, 27,  0,  0,  0,  0,  0,
250     20, 10,  0,  6, 15,  9,  0,  1,  0,  0,  0,
260     31, 18,  0, 10, 15,  0,  1, 13,  0,  0,  0}
270 m_vset(2,v)
280 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN      S.E.3
290 v={59, 15,  2,  1,200,127,  0,  7,  0,  3,  0,
300 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
310     31,  8,  1,  8,  7, 40,  2,  2,  5,  3,  0,
320     31,  8,  8,  7,  5, 36,  1,  2,  1,  1,  0,
330     31,  3,  7,  8,  1, 29,  1, 14,  3,  0,  0,
340     31,  0,  0,  9,  0,  2,  2,  8,  5,  2,  0}
350 m_vset(3,v)
360 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN      GLOCKEN
370 v={52, 15,  2,  1,200,127,  0,  0,  0,  3,  0,
380 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
390     26,  3,  0,  2, 15, 35,  1,  6,  3,  0,  0,
400     31,  6,  0,  6, 15,  0,  1,  2,  4,  0,  0,
410     31,  6,  0,  1, 14, 41,  1, 10,  7,  0,  0,
420     31,  7,  0,  6, 15,  0,  1,  2,  7,  0,  0}
430 m_vset(4,v)
440 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN      FM SYNTHE
450 v={44, 15,  2,  1,200,127,  0,  0,  0,  3,  0,
460 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
470     31,  0,  0,  0,  0, 22,  0,  2,  3,  0,  0,
480     27,  9,  0,  5,  1,  0,  0,  8,  3,  0,  0,
490     31,  0,  0,  0,  0, 23,  0,  4,  7,  0,  0,
500     27,  9,  0,  6,  1,  0,  0,  4,  7,  0,  0}
510 m_vset(5,v)
520 str a[256],b[256],c[256],d[256],e[256],f[256],g[256],h[256
],aa[256]
530 str j[256],k[256],l[256],m[256],n[256],o[256],p[256],q[256
],r[256]
540 str s[256],t[256],u[256],w[256],x[256],y[256],z[256]
550 str a1[256],b1[256],c1[256],d1[256],e1[256],f1[256],g1[256
],h1[256]
560 str a2[256],b2[256],c2[256],d2[256],e2[256],f2[256],g2[256
],h2[256]
570 str j1[256],k1[256],l1[256],m1[256],n1[256],o1[256],q1[256
],r1[256]
580 str j2[256],k2[256],l2[256],m2[256],n2[256],o2[256],q2[256
],r2[256]
590 str s1[256],u1[256],v1[256],w1[256],x1[256],y1[256],z1[256
]
600 str s2[256],u2[256],v2[256],w2[256],x2[256],y2[256],z2[256
]
610 str u3[256],u4[256],a0[256],b3[256],h3[256],in[256],sel[25
6]
620 dim str Q(4)[256],R(4)[256],S(4)[256],T(4)[256],U(4)[256],
V(4)[256]
630 sel=bnd("e",12,8192,16383)
640 for i=1 to 14:m_alloc(i,6000):m_assign(i,i):m_trk(i,"@99")
:next:m_alloc(11,12000):m_assign(11,11)
650 /*
660 key 2,"m_play()@M":key 12,"M_STOP()@M":key 19,"SAVE"+chr$(
34)+"RUN"+chr$(5):key 7,"m_trk("
670 m_tempo(182)
680 /* * * *   N O T I C E   * * *
690 /*EDIT PROG.
700 /*   プログラム中で使われている全ての音色の
710 /*   JOY STICK というパラメータを
720 /*       JOYSTICK       P+02 -> P+12
730 /*   の様に変更して下さい。
740 /*SEQUENCER
750 /*   MIDI CH 1,11,12,13,14,15,16,10
760 /*   PAN      Tr3=9:1  Tr4=1:9  Tr6=8:2 (Other Tr=5:5)
770 /*GLOBAL (DRUM KIT)
780 /*   DRUM KIT 3の"TOM 2"を全て"TOM 1"にする。
790 /*   OPEN HHとCLOSED HHのPANを共に9:1にする。
800 /*
810 /*         TRACK 1 (BASS)
820 /*
830 a0="@n1 @47 o2 q8 @m0 @b8192 @d0"
840 in="v12L8r4 @u120b-4f4<e-drc r>b-r4.a-4. r4agg-f4.& f1
850 a="v12@u120L8|:b-<b-a-gfgre-4<e-d-c>b-<cr>>b-4<b-a-gfgre-4
<e-d-c>d-<@u127d->|1@u120e-<@u127e->:|@u120e-16<@u127e-.>>
860 x=bnd("e-",12,8192,0)
870 b="@u120L8|:b-<b-a-gfgre-4<e-d-c>b-<cr>>b-4<b-a-gfg|1r4<e-
d-16c16>b-4<ce-&@L2"+x+"@b8192L8>b-&:|re-4<e-d-cd-e-r4>>
880 b1="b-<b-a-gfgre-4<e-d-c>d-<@u127d->@u120e-<@u127e-> t165
v14 r4a-2. t174{gge-e-}1t182
890 c="o2@u120L8|:b-<b-a-gfgre-4<e-d-c>b-<cr>>b-4<b-a-gfgre-4|
1<e-d-c>d-<@u127d->@u120e-<@u127e->:|@u120q4<e->@u120q8d<@u127q6
d>@u120q8d-<@u127q6d->@u120q8c<@u127q6c>
900 d="L8o2q8a-<a->a-<g-fe-fg-4>a-<a-g-f<d->fe- >a-4<a-g-fe-f>
a-4<a->g-g-f<q6f>q8g-<q6g-> q8a-<a->a-<g-fe-fg-4a-g->f<f<d->fe-
>a-4<a-g-fe-f>a-4<a->g-g-f<q6f>q8g-<q6g->
910 e="L8o2q8g<gcfrcfg >g<gcf>b-<b-c<c>> g<gcfrcfg >g<gcfb-afc
>g<gcfrcfg >g<gcf>b-<b-c<c>> g<gcfcfgf >g<gcfb-afc
920 g="L8o2q8b-<b-a-ga-b-r>b-4<b-a-ga-b-<e->a- >a-<a->g<g>e-<e
->rf b-<b->a-<a->g<ge-<e-> >b-<b-a-ga-b-r>b-4<b-a-ga-b-<e->a- >a
-<a->g<g>e-<e-b->b- a-<a->ggf<fe-<e-
930 j="L8o2q8a-a-<g-a->a-<a-g->a-4a-<g-a-e-fd-e- >b-<q6b->q8g<
q6g>q8a-<q6a->q8a<q6a> q8b-<b->g<g>a-<a->f<f d-<d->c<c>>b-<b->a-
<a-> f<f>g<g>a-<a->b-<b-> daa<d4>dad g<d>g<g4>g<dr
940 k="L8v13o2q8|:6r1:|r2..b-4<b-a->a-<q7gfa-q8b->
950 l="L8|:3 {rb-rb-}1 :| r4<b-16a-16ga-b-r4> |:3 {rb-rb-}1 :|
rb-a-ga-gfg
960 m="L8v13@u120o2q8b-<b-a-gfq2gfq6e-4<q2e-d-c>q7b-<cr>>q8b-4
<b-a-gfq2gfq6e-4<q2e-d-c>q7d-<d-16>d-16e-<e-16>e-16 >b-4<b-ra-gf
>f4<fe-de-f4>b-4<b-a-gr<d-e-4 >q8f<q6f>q8e<q6e>q8e-<q6e->q8d<q6d
>
970 n=">b-<b->a-16<a-16ga-frg e-<e->d-<d->a-<a-rg->> b-16<b-16
b-a->b-<g>b-<fg b-<'e-a-''fb-'@u90'f16b-''g16<c>'@u120'f4b-'>>b-
<b-> b-<b->b-16b-16<a-4ge-f e-<e->e-16e-16<c>>a-<a->gg- f<b-fa-4
gf>b-< d-<q6d->q8c<q6c>>q8b-<q6b->q8a-<q6a->
980 o="@n1 @49 o4 q8 v1 @u127 @m0 @b8192L8 @d0r1 b-&v2b-&v3b-&
v5b-&v6b-&v8b-&v10b-&v12b-& b-1& @m127 b-1& @m0b-1& b-2&b-&v11b-
&v9b-&v7b-& t166v5b-&v3b-&v2b-&v1b-r2 t176@47r1t182
990 x="r>gr2. r1 b-4fr<e-drc4.>gr<ferd
1000 m_trk(1,a0+in)
1010 m_trk(1,a)
1020 m_trk(1,b):m_trk(1,b1)
1030 m_trk(1,"v12"+c)
1040 m_trk(1,d)
1050 m_trk(1,e)
1060 m_trk(1,d)
1070 m_trk(1,g)
1080 m_trk(1,g)
1090 m_trk(1,j)
1100 m_trk(1,k)
1110 m_trk(1,l)
1120 m_trk(1,m)
1130 m_trk(1,n)
1140 m_trk(1,g)
1150 m_trk(1,g)
1160 m_trk(1,j)
1170 m_trk(1,o)
1180 m_trk(1,"v13"+c)
1190 m_trk(1,d)
1200 m_trk(1,e)
1210 m_trk(1,d)
1220 m_trk(1,g)
1230 m_trk(1,g)
1240 m_trk(1,"v12"+j)
1250 m_trk(1,x)
1260 /*
1270 /*         TRACK 2 (BRASS)
1280 /*
1290 a0="@n12 @3 q8 v7  @m0 @b8192 @d0"
1300 in="@u120L8o5r4 'c4fb-'>'cfb-'r'gb-<d>''gb-<c>'r'gb-<c>' r
'gb-<d>'r4.'g-1b-<ce->'@L72@d127'a-<ce-f>' r1@d0
1310 a="@u110L8o5|:'c1fb-' r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb-' r
1 :|
1320 b1="'c1fb-'r4r'ce-a-'r'ce-a-'r'ce-a-' <r4@L144'cfb-'> L4@n
11@3v09@u120'egb-<d>''egb-<c>''e-gb-<c>''e-gb-<d>'
1330 c="@n11 @3 q8 v08 @m0 @b8192 @u110o5L8r1 r1 r1 r2r@L72'gb-
<e->' r1 r1 L8r2..@d127@u99'e-a-' '>b-<e-''>gb-<'@L72'>a-<d-''e-
g'
1340 d="@d0@u99o4q7L8r1 r1 r2r'a-<d-f>'r4 r'a-<d-g->'r4'a-<d-f'
r4. r1 r1 r2..@d127q8g- d->a-<f4d->a-{a-b-<d-e-}4
1350 e="@d0@u99o6L8 r1 r2.q4'c4dfg' r2.'>cfb-<'q6'cfb-' q4r2.'e
-4g<c>' r1 r2r8'>cdfg<''c4dfg' r2..@d127>q8g fc4g4g16c16f16g16<c
1360 g="@76 q8 o5 v13 @u120 @d0 r1 r1 r2..L16a-f d1 r1 r1 r1 r2
b-ge-c>b-ge-c&
1370 h="c1 r1 r1 @22o5'a-4<ce-f'r2. r1 r1 r1 r1
1380 j="@n11@3q6o4v07@u105@d0|:L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g>
'r'a-<cdf>'r 'a-<ce-g>''a-<ce-a->'r'a-<cdf>'r@d127a-<c'df'@d0> |
1'a-4<ce-g>'q7'fb-<ce->''fb-<ce->'r@L72'fb-<cd' L2'ce-a-b-''cdfb
-' :| >'a1<cdg'@L144q8'dfa-b'@1r4
1390 k="@n12 @3 q8 v07"+a
1400 m="|:5r1:|@n11o5q4@u75L8r2d-e-4. r1 r1
1410 n="|:8r1:|"
1420 o="@n12 @3 q8 v08 @u110 @d0L8o5'c1fb-' r4.'ce-a-'r'ce-a-'r
'ce-a-' 'c1fb-' r1"+b1
1430 r="@d0@u99o4q7L8r1 r1 r2r'a-<d-f>'r4 r'a-<d-g->'r4'a-<d-f'
r4. r1 r1 r2..q8g- d->a-<f4d->a-{a-b-<d-e-}4
1440 w="@n11@3q6o4v07@u105@d0|:L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g>
'r'a-<cdf>'r 'a-<ce-g>''a-<ce-a->'r'a-<cdf>'r@d127a-<c'df'@d0> |
1'a-4<ce-g>'q7'fb-<ce->''fb-<ce->'r@L72'fb-<cd' L2'ce-a-b-''cdfb
-' :| >'a1<cdg'L8|:4r'dfa-b':|
1450 x="q8r'dfa-b'r2. r1 'c4fb-'>L8'cfb-'r'gb-<d>''gb-<c>'r<@L7
2'dg<c>'L8>'dg<c>'r'ga<ce>''f+a<cd>'r'dfa<c>'
1460 m_trk(2,a0+in)
1470 m_trk(2,a)
1480 m_trk(2,a):m_trk(2,b1)
1490 m_trk(2,c)
1500 m_trk(2,d)
1510 m_trk(2,e)
1520 m_trk(2,d)
1530 m_trk(2,g)
1540 m_trk(2,h)
1550 m_trk(2,j)
1560 m_trk(2,k)
1570 m_trk(2,a)
1580 m_trk(2,m)
1590 m_trk(2,n)
1600 m_trk(2,g)
1610 m_trk(2,h)
1620 m_trk(2,j)
1630 m_trk(2,o)
1640 m_trk(2,c)
1650 m_trk(2,r)
1660 m_trk(2,e)
1670 m_trk(2,r)
```

▶夏ですので，X68000ちゃんと海へ行ったらグレーのX68000ちゃんが真っ黒になって帰りました。
関口　敬文(15)兵庫県

```
 1680 m_trk(2,g)
 1690 m_trk(2,h)
 1700 m_trk(2,w)
 1710 m_trk(2,x)
 1720 /*
 1730 /*         TRACK 3 (PIANO)
 1740 /*
 1750 a0="@n13 @72 o5 q8 v09 @m0 @b8192 @d0"
 1760 in="@u99L8o5r4 'c4fb-'>r4'gb-<d>''gb-<c>'r'gb-<c>' r'gb-<d
>'r4.'b-1<ce->'@L72@d127'a-<ce-f>' r1@d0
 1770 a="@u115L8o5|:'c1fb-' r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb-' r
1 :|
 1780 b1="'c1fb-'r4r'ce-a-'r'ce-a-'r'ce-a-' <r4@L144'cfb-'> L4@n
12@52v07q6<'egb-<d>''egb-<c>''e-gb-<c>''e-gb-<d>'
 1790 c="L8|:'cfb-'rr'ce-a-'rr'>b-<e-a-'r |1r'e-gb-'rr'e-a-<c>'r
4. :| r'>b-<e-g'r4.'>g4b-<e-'r
 1800 c1="'cfb-'rr'ce-a-'rr'>b-<e-a-'r r'e-gb-'rr'e-a-<c>'r4. 'c
fb-'rr'ce-a-'r4.@d127a- e->b-<d-4.g4.
 1810 d="@d0'>b-<e-a-'rr'>b-<e-g'r4'>a-<d-f'r r'>a-4<d-g-'r'd-g-
b'r4. '>b-<e-a-'rr'>b-<e-g'r'>a-<d-f'r4 r'>a-<d-g-'r4'>a-<d-f'r4
.
 1820 d1="'>a-<e-a-'rr'>b-<e-g'r4'>a-<d-f'r r'>a-4<d-g-'r'd-g-b-
'r4. '>a-<e-a-'rr'>a-<e-g'r4.@d127g- d->a-<f4d->a-{a-b-<d-e-}4
 1830 e="@d0|:'>g<cf'rr'>g<ce'r4'cfb-'r |1r'c2fa''cdf'gr:| r'c4f
a'r4.'e-g<c>'r
 1840 e1="'>g<cf'rr'>g<ce'r4'cfb-'r r'c2fa''cdf'gr '>g<cf'rr'>g<
ce'r4.@d127g fc4g4g16c16f16g16<c>
 1850 g=">|:@d0@L72q8'cdfb-'@d127'>a-1b-<df'L8r|1@d0'fa-b-<e->''
fa-b-<d>'r@d127'e-1fa-<c>'rr1 :|@d0'fa-b-<e->''fa-b-<f>'r@L168'e
-gb-'L8r@d127'e-1gb-<c>'r
 1860 h="|:@d0@L72q8'cdfb-'@d127'>a-1b-<df'L8r|1@d0'fa-b-<e->''f
a-b-<d>'r@d127'e-1fa-<c>'rr1 :|@d0'fa-b-<e->''fa-b-<f>'r@L168'e-
gb-'r8@L144'e-gb-<c>'L8'a-<ce-g>'r'a-<ce-g>'
 1870 j="|:L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g>'r'a-<cdf>'r 'a-<ce-g
>''a-<ce-a->'r'a-<cdf>'r@d127a-<c'df'@d0> |1'a-4<ce-g>'q7'fb-<ce
->''fb-<ce->'r@L72'fb-<cd' L2'ce-a-b-''cdfb-' :| >'a1<cdg''d1fa-
b'
 1880 k="q8 v09o5"+a
 1890 m="@n11 @05 o4 q6 v08 @u110L8r'cdfb-'r4'cdfb-'r4@L72'ce-fa
-'L8'ce-fa-'r'ce-g''ce-fa-'r4 r'>a-<ce-g'rq1'>a-<ce-'q4'>a-<ce-g
'q1'c16e-fa-''>a-16<ce-fa-'q4'>a-<ce-g'q1'ce-fa-' q6'c4e-fa-'r@d
127L32dfgb-r2
 1900 m1="@d0L8q6r'cdfb-'r4'cdfb-'r4@L72'ce-fa-'L8'ce-fa-'r'ce-g
'@L72'ce-fa-' L8r'>a-<ce-g'r4'>a-<ce-g'r'ce-fa-''ce-fa-' @u90q2'
>b-4<dfb-'q1'>f16b-<df''>f16<f''>f4b-<df''>b-<dfa-'q8'>b-4<dfgb-
'@u110
 1910 n="q6r'cdfb-'r4q8'c4e-fa-'q2'ce-fa-''ce-fa-' q4'>a-4<ce-g'
'c4e-fa-'q4'c4dfb-'r4 rq1'd32e-gb-'r32'd32b-'r32q4'de-gb-''de-gb
-'r'de-gb-'r'de-gb-' q6'c4e-fa-'r8q8@d127L32cdfb-r2
 1920 n1="@d0L8q6r'c4dfb-'rq4'cdfb-'r4@L72'ce-fa-'L8'ce-fa-'r'ce
-g'q6@L72'ce-fa-' L4q7'ce-fb-''ce-g''ce-fa-''ce-fa' L8@d127'b->b
-<''f>f<''e->e-<''d2>d<'"        /*r8
 1930 o="q8 o5 @u115 @d0 L8'c1fb-' r4.'ce-a-'r'ce-a-'r'ce-a-' 'c
1fb-' r1 "+b1
 1940 r1="'>a-<e-a-'rr'>b-<e-g'r4'>a-<d-f'r r'>a-4<d-g-'r'd-g-b-
'r4. '>a-<e-a-'rr'>a-<e-g'r4.g- d->a-<f4d->a-{a-b-<d-e-}4
 1950 w="|:L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g>'r'a-<cdf>'r 'a-<ce-g
>''a-<ce-a->'r'a-<cdf>'r@d127a-<c'df'@d0> |1'a-4<ce-g>'q7'fb-<ce
->''fb-<ce->'r@L72'fb-<cd' L2'ce-a-b-''cdfb-' :| >'a1<cdg'L8|:4r
'dfa-b':|
 1960 m_trk(3,a0+in)
 1970 m_trk(3,a)
 1980 m_trk(3,a):m_trk(3,b1)
 1990 m_trk(3,"@n12     o6 q7 v07 @m0 @b8192 @u115"+c):m_trk(3,c
1)
 2000 m_trk(3,d):m_trk(3,d1)
 2010 m_trk(3,e):m_trk(3,e1)
 2020 m_trk(3,d):m_trk(3,d1)
 2030 m_trk(3,g)
 2040 m_trk(3,h)
 2050 m_trk(3,j)
 2060 m_trk(3,"@n13 @72"+k)
 2070 m_trk(3,a)
 2080 m_trk(3,m):m_trk(3,m1)
 2090 m_trk(3,n):m_trk(3,n1)
 2100 m_trk(3,"@n12 @52 o6 q7 v07 @b8192 @u115 r8"+g)
 2110 m_trk(3,h)
 2120 m_trk(3,j)
 2130 m_trk(3,"v09"+o)
 2140 m_trk(3,"@n12 o6 q7 v7 @m0 @b8192 @u115"+c):m_trk(3,c1)
 2150 m_trk(3,d):m_trk(3,r1)
 2160 m_trk(3,e):m_trk(3,e1)
 2170 m_trk(3,d):m_trk(3,r1)
 2180 m_trk(3,g)
 2190 m_trk(3,h)
 2200 m_trk(3,w)
 2210 m_trk(3,x)
 2220 /*       MT-32
 2230 m_trk(6,"@n2 @4 o5 q7 v06 @m0 @b8292 @u90 p1"+in)
 2240 m_trk(6,"|:20r1:|")
 2250 m_trk(6,"o5"+c):m_trk(6,c1)
 2260 m_trk(6,d):m_trk(6,d1)
 2270 m_trk(6,e):m_trk(6,e1)
 2280 m_trk(6,d):m_trk(6,d1)
 2290 m_trk(6,g)
 2300 m_trk(6,h)
 2310 m_trk(6,"p2"+j)
 2320 m_trk(6,"v6 o5 @p10"+a)
 2330 m_trk(6,a)
 2340 m_trk(6,"|:16r1:|")
 2350 m_trk(6,"o5 q7 v06 @b8292 @u90 p1"+g)
 2360 m_trk(6,h)
 2370 m_trk(6,"p2"+j+"p1")
 2380 m_trk(6,o)
 2390 m_trk(6,"@n2 o5 q7 v06 @b8292 @u90 p1"+c):m_trk(6,c1)
 2400 m_trk(6,d):m_trk(6,d1)
 2410 m_trk(6,e):m_trk(6,e1)
 2420 m_trk(6,d):m_trk(6,d1)
 2430 m_trk(6,"o5 q7 v06 @b8292 @u90 p1"+g)
 2440 m_trk(6,h)
 2450 m_trk(6,"p2"+w+"p1")
 2460 m_trk(6,x)
 2470 /*
 2480 /*         TRACK 4 (PIANO 2)
 2490 /*
 2500 a0="@n14 @3 o5 q8 v07 @m0 @d0
 2510 in="r4 |:4r1:|"
 2520 a="@b8292@u110L8|:'c1fb-' r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb
-' r1 :|
 2530 b1="'c1fb-'r4r'ce-a-'r'ce-a-'r'ce-a-' <r4@L144'cfb-'> L4@n
13@72v08q6'egb-<d>''egb-<c>''e-gb-<c>''e-gb-<d>'
 2540 c="L8|:r'cfb-'r4'ce-a-'rr'>b-<e-a-' |1r4'e-gb-'rr'e-a-<c>'
r4 :| r4'>b-4<e-g'r4'>g4b-<e-'
 2550 c1="r'cfb-'r4'ce-a-'rr'>b-<e-a-' r4'e-gb-'rr'e-a-<c>'r4 r'
cfb-'r4'ce-a-'r4@d127a- e->b-<d-4.g4.
 2560 d="@d0r'>b-<e-a-'rr'>b-<e-g'r4'>a-<d-f' r4'>a-4<d-g-'r'd-g
-b'r4 r'>b-<e-a-'rr'>b-<e-g'r4'>a-<d-f' r4'>a-<d-g-'r4'>a-<d-f'r
4
 2570 d1="r'>a-<e-a-'rr'>b-<e-g'r4'>a-<d-f' r4'>a-4<d-g-'r'd-g-b
-'r4 r'>a-<e-a-'rr'>a-<e-g'r4@d127g- d->a-<f4d->a-{a-b-<d-e-}4
 2580 e="@d0|:r'>g<cf'r4'>g<ce'rr'cfb-' |1r4'c4fa'r'cdf'gr:| r4'
c4fa'r4.'e-g<c>'
 2590 e1="r'>g<cf'r4'>g<ce'rr'cfb-' r4'c4fa'r'cdf'gr r'>g<cf'r4'
>g<ce'r4@d127g fc4g4g16c16f16g16<c>
 2600 g="@b8292|:@d0@L72q8'cdfb-'@d127'>a-1b-<df'L8r|1@d0'fa-b-<
e->''fa-b-<d>'r@d127'e-1fa-<c>'rr1 :|@d0'fa-b-<e->''fa-b-<f>'r@L
168'e-gb-'L8r@d127'e-1gb-<c>'r
 2610 h="|:@d0@L72q8'cdfb-'@d127'>a-1b-<df'L8r|1@d0'fa-b-<e->''f
a-b-<d>'r@d127'e-1fa-<c>'r@d0'a-1<ce-f>' :|@d0'fa-b-<e->''fa-b-<
f>'r@L168'e-gb-'r8@L144'e-gb-<c>'L8'a-<ce-g>'r'a-<ce-g>'
 2620 j="@b8192|:8 r1 :|
 2630 k="q8 v07o5"+a
 2640 m="@n12 @05 o4 q6 v07 @u90 @b8292L8r32r'cdfb-'r4'cdfb-'r4@
L72'ce-fa-'L8'ce-fa-'r'ce-g''ce-fa-'r4 r'>a-<ce-g'rq1'>a-<ce-'q4
'>a-<ce-g'q1'c16e-fa-''>a-16<ce-fa-'q4'>a-<ce-g'q1'ce-fa-' q6'c4
e-fa-'r@d127L32dfgb-r2
 2650 m1="@d0L8q6r'cdfb-'r4'cdfb-'r4@L72'ce-fa-'L8'ce-fa-'r'ce-g
'@L72'ce-fa-' L8r'>a-<ce-g'r4'>a-<ce-g'r'ce-fa-''ce-fa-' @u70q2'
>b-4<dfb-'q1'>f16b-<df''>f16<f''>f4b-<df''>b-<dfa-'q8'>b-4<dfgb-
'@u90
 2660 o="@n13 @3 q8 v07 o5@b8292@u110@d0L8'c1fb-' r4.'ce-a-'r'ce
-a-'r'ce-a-' 'c1fb-' r1"+b1
 2670 r1="r'>a-<e-a-'rr'>b-<e-g'r4'>a-<d-f' r4'>a-4<d-g-'r'd-g-b
-'r4 r'>a-<e-a-'rr'>a-<e-g'r4g- d->a-<f4d->a-{a-b-<d-e-}4
 2680 x="<q8r1 r1 'c4fb-'>L8'cfb-'r'gb-<d>''gb-<c>'r<@L72'dg<c>'
L8>'dg<c>'r'ga<ce>''f+a<cd>'r'dfa<c>'
 2690 m_trk(4,a0+in+a)
 2700 m_trk(4,a):m_trk(4,b1)
 2710 m_trk(4,"@n13     o5 q7 v08 @m0 @b8192 @u115"+c):m_trk(4,c
1)
 2720 m_trk(4,d):m_trk(4,d1)
 2730 m_trk(4,e):m_trk(4,e1)
 2740 m_trk(4,d):m_trk(4,d1)
 2750 m_trk(4,g)
 2760 m_trk(4,h)
 2770 m_trk(4,j)
 2780 m_trk(4,"@n14 @3"+k)
 2790 m_trk(4,a)
 2800 m_trk(4,m):m_trk(4,m1)
 2810 m_trk(4,n):m_trk(4,n1)
 2820 m_trk(4,"@n13 @72 o5 q7 v08 @u115 r16."+g)
 2830 m_trk(4,h)
 2840 m_trk(4,j)
 2850 m_trk(4,o)
 2860 m_trk(4,"@n13 @72 o5 q7 v08 @m0 @b8192 @u115"+c):m_trk(4,c
1)
 2870 m_trk(4,d):m_trk(4,r1)
 2880 m_trk(4,e):m_trk(4,e1)
 2890 m_trk(4,d):m_trk(4,r1)
 2900 m_trk(4,g)
 2910 m_trk(4,h)
 2920 m_trk(4,j)
 2930 m_trk(4,j)
 2940 /*        MT-32
 2950 m_trk(7,in+"|:20r1:|")
 2960 m_trk(7,"@n3 @1 o4 q7 v08 @m0 @b8292 @u99 p2"+c):m_trk(7,c
1)
 2970 m_trk(7,d):m_trk(7,d1)
 2980 m_trk(7,e):m_trk(7,e1)
 2990 m_trk(7,d):m_trk(7,d1)
 3000 m_trk(7,g)
 3010 m_trk(7,h)
 3020 m_trk(7,j)
 3030 m_trk(7,"|:32r1:|")
 3040 m_trk(7,"o4 q7 @b8292 @u99 p2"+g)
 3050 m_trk(7,h)
 3060 m_trk(7,j)
 3070 m_trk(7,"|:8r1:|")
 3080 m_trk(7,"@n3 o4 q7 v08 @b8292 @u99 p2"+c):m_trk(7,c1)
 3090 m_trk(7,d):m_trk(7,d1)
 3100 m_trk(7,e):m_trk(7,e1)
 3110 m_trk(7,d):m_trk(7,d1)
 3120 m_trk(7,g)
 3130 m_trk(7,h)
 3140 m_trk(7,j)
 3150 m_trk(7,x)
 3160 /*
 3170 /*         TRACK 5 (BRASS 2)
 3180 /*
 3190 a0="@n15 @73 q8 v11 @m0 @b8292 @d0"
 3200 in="@u120L8o6r4 'c4fb-'>'cfb-'r'gb-<d>''gb-<c>'r'gb-<c>' r
'gb-<d>'r4.'g-1b-<ce->'@L72@d127'a-<ce-f>' r1@d0
 3210 a="@u110L8o6|:'c1fb-' r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb-' r
1 :|
 3220 b1="'c1fb-'r4r'ce-a-'r'ce-a-'r'ce-a-' <r4@L144'cfb-'> r1
 3230 c="@n15 @73 q8 v11 @m0 @b8292 @u110o6L8r1 r1 r1 r2r@L72'gb
-<e->' r1 r1 L8r2..@d127@u99'e-a-' '>b-<e-''>gb-<'@L72'>a-<d-''e
-g'
 3240 d="@d0@u99o5q7L8r1 r1 r2r'a-<d-f>'r4 r'a-<d-g->'r4'a-<d-f'
r4. r1 r1 r2..@d127q8g- d->a-<f4d->a-{a-b-<d-e-}4
 3250 e="@d0@u99o7L8 r1 r2.q4'c4dfg' r2.'>cfb-<'q6'cfb-' q4r2.'c
4e-g' r1 r2r8'>cdfg<''c4dfg' r2..@d127>q8g fc4g4g16c16f16g16<c
```


▶明日の「Oh!X」のためにがんばって！　夜ふかししちゃだめ！　たばこもすいすぎはだめ！　お酒もほどほどに！　以上。
土田　泰正(16)大阪府

```
 3260 g="@76 q8 o5 v13 @u120 @d0 @b8292r1 r1 r2..L16r32a-f d1 r1
 r1 r1 r2b-ge-c>b-ge-c32&
 3270 h="c1|:7 r1 :|"
 3280 j="@n15@73q6o5v11@u105@d0|:L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g
>'r'a-<cdf>'r 'a-<ce-g>''a-<ce-a->'r'a-<cdf>'r@d127a-<c'df'@d0>
|1'a-4<ce-g>'q7'fb-<ce->''fb-<ce->'r@L72'fb-<cd' L2'ce-a-b-''cdf
b-' :| >'a1<cdg''d1fa-b'
 3290 k="q8 v11"+a
 3300 m="|:8r1:|
 3310 o="@n11@49 o4 q8 v1 @u127 @m0 @b8292L8 @d0r1 b-&v2b-&v3b-&
v4b-&v6b-&v8b-&v10b-&v12b-& b-1& @m127 b-1& @m0b-1& b-2&b-&v11b-
&v9b-&v7b-& v5b-&v3b-&v2b-&v1b-v11r2 r1
 3320 solo2()
 3330 w="@n15@73q6o5v11@u105@d0|:L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g
>'r'a-<cdf>'r 'a-<ce-g>''a-<ce-a->'r'a-<cdf>'r@d127a-<c'df'@d0>
|1'a-4<ce-g>'q7'fb-<ce->''fb-<ce->'r@L72'fb-<cd' L2'ce-a-b-''cdf
b-' :| >'a1<cdg'L8|:4r'dfa-b':|
 3340 x="q8r'dfa-b'r2. r1 'c4fb-'>L8'cfb-'r'gb-<d>''gb-<c>'r<@L7
2'dg<c>'L8>'dg<c>'r'ga<ce>''f+a<cd>'r'dfa<c>'
 3350 m_trk(5,a0+in)
 3360 m_trk(5,a)
 3370 m_trk(5,a):m_trk(5,b1)
 3380 m_trk(5,c)
 3390 m_trk(5,d)
 3400 m_trk(5,e)
 3410 m_trk(5,d)
 3420 m_trk(5,g)
 3430 m_trk(5,h)
 3440 m_trk(5,j)
 3450 m_trk(5,k)
 3460 m_trk(5,a)
 3470 m_trk(5,m)
 3480 m_trk(5,m)
 3490 m_trk(5,g)
 3500 m_trk(5,h)
 3510 m_trk(5,j)
 3520 m_trk(5,o)
 3530 m_trk(5,c)
 3540 for i=0 to 4:m_trk(5,R(i)).next
 3550 m_trk(5,"@73 q8 v11 @b8292@u110"+e)
 3560 for i=0 to 4:m_trk(5,T(i)):next
 3570 for i=0 to 4:m_trk(5,U(i)):next
 3580 for i=0 to 4:m_trk(5,U(i)):next
 3590 m_trk(5,w)
 3600 m_trk(5,x)
 3610 /*
 3620 /*         TRACK 8 (PERCUSSIONS)
 3630 /*
 3640 a0="@n16 @50 q8 @v127 @m0 @b8192 @d0"
 3650 in="r4 @u110o6L4 ccc8c8r8c8 r8c8r.c8r r2r8c. r2.c
 3660 a="@u110o6L4|:7crcr:|crcc
 3670 b="|:3crcr:|r2c8r. |:4crcr:|
 3680 b1="crcr crcc rcr2 cccc
 3690 c="@u110o6L4|:8crcr:|
 3700 g="L4|:8@u110o6c.@u120o3e8@u110o6c.@u120o3e8:|
 3710 h="L4|:7@u110o6c.@u120o3e8@u110o6c.@u120o3e8:|@u110o6c.@u1
20o3e4@u110o6c8c
 3720 j="@u110o6L4|:7crcr:|cr.c8c
 3730 k="@u110o6L4|:8crcr:|
 3740 l="|:3crcr:|c.c8r2 |:3crcr:| cr.c.
 3750 m="@u110o6L4|:7crcr:|crc.c8
 3760 o="@u110o6L4|:6cccc:| r1 ccr.c8
 3770 w="@u110o6L4|:7crcr:|cccc
 3780 x="cr2. r1 o3'd<<<c>>>'d'd8<<<c>>>'d8r8o6c8 rc{ccrc}2
 3790 m_trk(8,a0+in)
 3800 m_trk(8,a)
 3810 m_trk(8,b):m_trk(8,b1)
 3820 m_trk(8,c)
 3830 m_trk(8,c)
 3840 m_trk(8,c)
 3850 m_trk(8,c)
 3860 m_trk(8,g)
 3870 m_trk(8,h)
 3880 m_trk(8,j)
 3890 m_trk(8,k)
 3900 m_trk(8,l)
 3910 m_trk(8,m)
 3920 m_trk(8,m)
 3930 m_trk(8,g)
 3940 m_trk(8,h)
 3950 m_trk(8,j)
 3960 m_trk(8,o)
 3970 m_trk(8,c)
 3980 m_trk(8,c)
 3990 m_trk(8,c)
 4000 m_trk(8,c)
 4010 m_trk(8,g)
 4020 m_trk(8,h)
 4030 m_trk(8,w)
 4040 m_trk(8,x)
 4050 /*
 4060 a0="@n16 q8"
 4070 in="o6L4@u99e16@u127e8. r1 r1 @u110o3r4c8>a8f8r4. r2.o6@L8
r@u99e@u110e @u115e@u120e@u127e
 4080 a="@u127o6L4|:7rere:|re.e8.e8.
 4090 b="|:3rere:| @u110o3c8>a16a16fr8@u127o6e. |:3rere:|re.e8r
 4100 b1="rere rer{reee}4 er2. r1
 4110 c="@u127o6L4|:7rere:| re.e8e
 4120 d="@u127o6L4|:8rere:|
 4130 e="@u127o6L4|:7rere:| r8e8@u120eL8r@u120e@u127ee
 4140 f="@u127o6L4|:7rere:| r8@u110o3c8c4>f8f8o6@u120e16e8e16
 4150 g="@u127o6L4|:7rere:| rer8e8e8r8
 4160 h="@u127o6L4|:7rere:| r8e8e8r8r8e8{eere}4
 4170 j="@u127o6L4|:3rere:| |:@u110o3d@u127o6e:| |:rere:| @u110o
3d@u127o6ere @u110o3d@u127o6ee8@u110o2f16@u127o6e16r8e16e16
 4180 k="@u127o6L4|:6rere:| rer8e8e r8e8er@u110o2f8r8
 4190 l="@u127o6L4|:3rere:| L8ree16e16reer4 |:3r4e4r4e4:| @u110o
3rcc>fr4@u127o6e4
 4200 m="@u127o6L4|:7rere:| re@u110o3r16c16c8>f8f8
 4210 n="@u127o6L4|:7rere:| re@u110o3c16c16>a8a16f16f8
 4220 o="@u120o3L8>far4f4.<c16c16 c>aa4.fr4 rar4f4.<c16c16 rcc4>
r16a16af4 rar4f4.<c16c16 c>aa4.fr4 @u127o6e4e4r2 r2e4e4
 4230 w="@u127o6L4|:3rere:| |:@u110o3d@u127o6e:| |:rere:| @u110o
3d@u127o6ere L8@u120o3d@u127o6e|:3re:|
 4240 x="@u127r'e>>>d'r2. o3@u99c16c16@u110cc>aaffo6@u127e r4e4o
2f'f<<<<e'ro3@u110d& d4d4ddrd& d1
 4250 m_trk(9,a0+in)
 4260 m_trk(9,a)
 4270 m_trk(9,b):m_trk(9,b1)
 4280 m_trk(9,c)
 4290 m_trk(9,d)
 4300 m_trk(9,e)
 4310 m_trk(9,f)
 4320 m_trk(9,g)
 4330 m_trk(9,h)
 4340 m_trk(9,j)
 4350 m_trk(9,k)
 4360 m_trk(9,l)
 4370 m_trk(9,m)
 4380 m_trk(9,n)
 4390 m_trk(9,g)
 4400 m_trk(9,h)
 4410 m_trk(9,j)
 4420 m_trk(9,o)
 4430 m_trk(9,c)
 4440 m_trk(9,d)
 4450 m_trk(9,e)
 4460 m_trk(9,f)
 4470 m_trk(9,g)
 4480 m_trk(9,h)
 4490 m_trk(9,w)
 4500 m_trk(9,x)
 4510 /*
 4520 a0="@n16 q8"
 4530 in="@u99o3L4r4 ddd8d8r8d8 r8d4.r8@u127d4. r2r8d4.& d1
 4540 a="@u110o2L8|:3f+f+a+f+f+a+f+f+:|f+f+a+f+f+a+f+a+ 'f+<d>'f
+a+f+f+a+f+f+|:f+f+a+f+f+a+f+f+:|f+f+a+f+f+<d>f+16<@u120d.>@u110
 4550 b="'f+<d>'f+a+f+f+a+f+f+|:f+f+a+f+f+a+f+f+:|r2f+<@u120d4.>
@u110'f+<d>'f+a+f+f+a+f+f+|:f+f+a+f+f+a+f+f+:| {f+rf+r}1
 4560 b1="'f+<d>'f+a+f+f+a+f+f+ {f+a+f+a+}1 @u120<d1  L4|:@u110d
@u120d:|
 4570 c="@u110o2L8'f+<d>'f+a+f+f+a+f+f+ |:3f+f+a+f+f+a+f+a+ f+f+
a+f+f+a+f+f+:| L4f+a+f+a+
 4580 d="@u110o2L8'f+<d>'f+a+f+f+a+f+f+ |:3f+f+a+f+f+a+f+a+ f+f+
a+f+f+a+f+f+:| L4'f+<d>'a+f+a+
 4590 e="@u110o2L8|:3f+f+a+f+f+a+f+f+ f+f+a+f+f+a+f+a+ :| f+f+a+
f+f+a+f+f+ L4f+a+f+a+8<d8>
 4600 f="@u110o2L8'f+<d>'f+a+f+f+a+f+f+ |:3f+f+a+f+f+a+f+a+ f+f+
a+f+f+a+f+f+:| L4f+a+f+a+8<d8>
 4610 g="@u110o2L8'f+<d>'a+f+f+|:15f+a+f+f+:|
 4620 j="@u110o2L8'f+<d>'f+a+f+f+a+f+f+ |:3f+f+a+f+f+a+f+a+ f+f+
a+f+f+a+f+f+:| L4f+a+f+a+
 4630 k="@u110o3L8d1> r1 r1 r2f+f+16f+16f+a+ |:3r1:| r2f+f+16f+1
6f+a+
 4640 l="<{rdrd}1 r4d4>f+f+16f+16'f+<d>'a+ <{rdrd}1> r4<d4r4>f+f
+16f+16 <|:3{rdrd}1:| r1
 4650 m="@u110o2L8|:56f+:|f+16f+16a+f+a+r2
 4660 o="@u110o2L4|:24f+:| f+@u120'f+<d'r2 ddr2
 4670 m_trk(10,a0+in)
 4680 m_trk(10,a)
 4690 m_trk(10,b):m_trk(10,b1)
 4700 m_trk(10,c)
 4710 m_trk(10,d)
 4720 m_trk(10,e)
 4730 m_trk(10,f)
 4740 m_trk(10,g)
 4750 m_trk(10,g)
 4760 m_trk(10,j)
 4770 m_trk(10,k)
 4780 m_trk(10,l)
 4790 m_trk(10,m)
 4800 m_trk(10,m)
 4810 m_trk(10,g)
 4820 m_trk(10,g)
 4830 m_trk(10,j)
 4840 m_trk(10,o)
 4850 m_trk(10,c)
 4860 m_trk(10,d)
 4870 m_trk(10,e)
 4880 m_trk(10,f)
 4890 m_trk(10,g)
 4900 m_trk(10,g)
 4910 m_trk(10,j)
 4920 /*
 4930 /*         TRACK 11 (SE おまけです。)
 4940 /*
 4950 a0="@n17@1o3v12@u85q8@m0@b8192p3"
 4960 a="@L1|:2 r2."+se1+"r8. r2."+se1+"r16":a1=se1+"r16 r2."+se
1+"r8.":a2="r2r8"+se1+"r8."+se1+"r16:|"
 4970 b="|:2 r2."+se1+"r8. r2."+se1+"r16":b1=se1+"r16 r2."+se1+"
r8. r1:|
 4980 b2="r2."+se1+"r8. r2."+se1+"r16":b3=se1+"r16r1r1"
 4990 k="|:7L16@2o4q8v12@u115@b8192 p3<er>er<er>er r8@1v12@u95o3
p3
 5000 x=bnd("e",12,8192,15705)
 5010 k1="@L1"+x+"r8."+x+"r16"
 5020 k2="@2@v127@u127@b8192o4p3e8p2c8>a4@3o2v11@u80@L3p3e&f&f+&
g&g+&a&a+&b&<c&c+&d&d+&p1e&f&f+&g& @u70g+&a&a+&b&<c&c+&d&d+&@u60
e&e+&f&g&g+&a&a+&b :|
 5030 m="|:8r1:|
 5040 o="|:3L16@2o4q8v12@u115@b8192 p3<er>er<er>er r8@1v12@u95o3
p3
 5050 m_trk(11,a0+"r4 |:4r1:|")
 5060 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5070 m_trk(11,b):m_trk(11,b1):m_trk(11,b2):m_trk(11,b3)
 5080 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5090 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5100 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5110 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5120 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5130 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5140 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
```

▶♪ぶーん　てけてけて　ぶーん　てけてけて　なかよし　よーいこ　みなおいで♪　ポンキッキは忘れかけていたものを思い出させてくれるとてもよい番組だ。

内海　雄介(17)石川県

```
 5150 m_trk(11,k):m_trk(11,k1):m_trk(11,k2):m_trk(11,k):m_trk(11
,k1):m_trk(11,"r1")
 5160 m_trk(11,m)
 5170 m_trk(11,m)
 5180 m_trk(11,a0)
 5190 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5200 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5210 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5220 m_trk(11,o):m_trk(11,k1):m_trk(11,k2):m_trk(11,k):m_trk(11
,k1):m_trk(11,"r1")
 5230 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5240 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5250 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5260 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5270 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5280 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5290 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
 5300 /*
 5310 /*         TRACK 12 13 14 (FM SYNTHESIZER)
 5320 /*
 5330 a0="@n18@m0
 5340 in="r4 r1 r2.@4o3q8v10@u50@b8192p3L16a-<ce-f e-fa-<c>a-<ce
-fe-fa-<c>a-<ce-f e-fa-<c>a-<ce-fe-fa-<c>a-<ce-f @5o5c2&@m30c2@m
0 |:7r1:|
 5350 a="|:8r1:|
 5360 b="|:12r1:|
 5370 g="L16r1 r1 r2..@4o5q8v10@u60@b8192p2a-f d1 r1 r1 r1 r2L16
b-ge-c>b-ge-c
 5380 h="r1 r1 r1 'a-1<ce-f' r1 r1 r1 r1
 5390 j="|:8r1:|"
 5400 n="|:32r1:|"
 5410 x=bnd("a-",12,8192,6143)
 5420 o="@5q8@b8192p3 r1 L8v11o2@u10b-&@u20b-&@u30b-&@u40b-&@u50
b-&@u60b-4.& b-1& @m127 b-1& @m0 b-1& b-2L8o4v12@u80p2"+x+"& @m3
0a-2&@m0a-2
 5430 solo(0)
 5440 r="|:8r1:|"
 5450 m_trk(12,a0+in)
 5460 m_trk(12,b)
 5470 m_trk(12,a+a+a+a)
 5480 m_trk(12,g)
 5490 m_trk(12,h)
 5500 m_trk(12,j+n)
 5510 m_trk(12,g)
 5520 m_trk(12,h)
 5530 m_trk(12,j)
 5540 m_trk(12,o)
 5550 m_trk(12,"   @5 q8 o4 v11 @u120 p2"+Q(0)):for i=1 to 4:m_t
rk(12,Q(i)):next
 5560 m_trk(12,r)
 5570 for i=0 to 3:m_trk(12,S(i)):next
 5580 m_trk(12,r)
 5590 for i=0 to 3:m_trk(12,"@u115"+U(i)):next
 5600 for i=0 to 2:m_trk(12,V(i)):next
 5610 /*
 5620 a0="@n19@m0"
 5630 in="r16r4 r1 r2.@4o3q8v10@u40@b8242p3L16a-<ce-f e-fa-<c>a-
<ce-fe-fa-<c>a-<ce-f e-fa-<c>a-<ce-fe-fa-<c>a-<ce-f @5o5f2&@m30f
4..@m0 |:7r1:|
 5640 g="L16r1 r1 r2..r32@4o5q8v10@u50@b8242p2a-f d2&d4... r1 r1
 r1 r2L16r32b-ge-c>b-ge-c32
 5650 x=bnd("a-",12,8242,6193)
 5660 o="@5q8@b8242p3 r1 L8v10o2@u10b-&@u20b-&@u30b-&@u40b-&@u50
b-&@u60b-4.& b-1& @m127 b-1& @m0 b-1& b-2L8o4v11@u80p2"+x+"& @m3
0a-2&@m0a-2
 5670 solo(50)
 5680 m_trk(13,a0+in)
 5690 m_trk(13,b)
 5700 m_trk(13,a+a+a+a)
 5710 m_trk(13,g)
 5720 m_trk(13,h)
 5730 m_trk(13,j+n)
 5740 m_trk(13,g)
 5750 m_trk(13,h)
 5760 m_trk(13,j)
 5770 m_trk(13,o)
 5780 m_trk(13,"r16@5 q8 o4 v10 @u99  p2"+Q(0)):for i=1 to 4:m_t
rk(13,Q(i)):next
 5790 m_trk(13,r)
 5800 for i=0 to 3:m_trk(13,S(i)):next
 5810 m_trk(13,r)
 5820 for i=0 to 3:m_trk(13,"@u95"+U(i)):next
 5830 for i=0 to 2:m_trk(13,V(i)):next
 5840 /*
 5850 a0="@n20@m0"
 5860 in="r8r4 r1 r2.@4o3q8v10@u40@b8292p2L16a-<ce-f e-fa-<c>a-<
ce-fe-fa-<c>a-<ce-f e-fa-<c>a-<ce-fe-fa-<c>a-<ce-f @5o5b-2&@m30b
-4.@m0 |:7r1:|
 5870 g="L16r1 r1 r2...@4o5q8v10@u40@b8292p2a-f d2... r1 r1 r1 r
2L16rp2b-ge-c>b-ge-
 5880 x=bnd("a-",12,8292,6243)
 5890 o="@5q8@b8292p3 r1 L8v09o2r16@u10b-&@u20b-&@u30b-&@u40b-&@
u50b-&@u60b-4.& b-1& @m127 b-1& @m0 b-1& b-2L8o4v10@u80p3"+x+"&
@m30a-2&@m0a-4..
 5900 solo(100)
 5910 m_trk(14,a0+in)
 5920 m_trk(14,b)
 5930 m_trk(14,a+a+a+a)
 5940 m_trk(14,g)
 5950 m_trk(14,h)
 5960 m_trk(14,j+n)
 5970 m_trk(14,g)
 5980 m_trk(14,h)
 5990 m_trk(14,j)
 6000 m_trk(14,o)
 6010 m_trk(14,"   @5 q8 o4 v10 @u99  p3"+Q(0)):for i=1 to 4:m_t
rk(14,Q(i)):next
 6020 m_trk(14,r)
 6030 for i=0 to 3:m_trk(14,S(i)):next
 6040 m_trk(14,r)
 6050 for i=0 to 3:m_trk(14,"@u95"+U(i)):next
 6060 for i=0 to 2:m_trk(14,V(i)):next
 6070 m_play():end
 6080 /*         E A S Y   B E N D   R O U T I N E by Z.N
 6090 func str bnd(A;str,L;float,V1;float,V2;float)
 6100 str B[256]
 6110 int I
 6120 float VL,V
 6130 VL=(V2-V1)/(L-1):B="":V=V1
 6140 for I=1 to L
 6150 if V>16383 then V=16383 else if V<0 then V=0
 6160 B=B+"@b"+str$(int(V))+A
 6170 V=V+VL
 6180 if I<>L then B=B+"&"
 6190 next
 6200 return(B)
 6210 endfunc
 6220 func fre()
 6230 int i
 6240 for i=1 to 14:print i;m_free(i):next
 6250 endfunc
 6260 func solo(a;int)
 6270 str s[256],t[256],u[256],v[256],w[256],x[256],y[256],z[256
]
 6280 u=bnd("b-",6,6826+a,8192+a):v=bnd("a-",6,6826+a,8192+a):w=
bnd("d",6,6826+a,8192+a)
 6290 Q(0)="@d0@m0@L2"+u+"&L8b-.&@m40b-4@m0a-gr@L2"+v+"&L8a-16&@
m40a-@m0b-rfre-r@L2"+w+"&L8d16&
 6300 x=bnd("g",6,6826+a,8192+a):y=bnd("c",6,6826+a,8192+a)
 6310 Q(1)="@m40d@m0e-fc>b-@L2"+v+"&L8a-16&@m40a-@m0@L2"+x+"&L8g
16&@m40g<@m0e-d@L2"+y+"&L8c16&@m40c
 6320 z=bnd("f",6,6826+a,8192+a)
 6330 Q(2)="@m0@L2>"+z+"&L8f16&@m40f@m0@L2"+u+"&L8b-16&@m40b-@m0
g<c>g<d>g<e-@L2"+z+"&f16&
 6340 Q(3)="@m40f4@m0L8e-dc@L2"+x+"&L8g16&@m40g@m0c@L2"+v+"&L8a-
16&@m40a-@m0c@L2"+u+"&L8b-16&@m40b-@m0c<c
 6350 z=bnd("d",12,8192+a,6826+a)
 6360 Q(4)="@L2"+w+"&L8d16& @m20d&@m30d&@L4@m40"+z+"r2>@m0@b"+st
r$(8192+a)
 6370 /*
 6380 /*
 6390 S(0)="L16r8fgb-agfagfe-gfe-d fe-dce-dcL8>b-.<cd>@L2"+x+"&L
8g16&@m40g@m0ab-<c>ab-<f>a
 6400 x=bnd("a",6,6826+a,8192+a)
 6410 S(1)="b-<@L2"+y+"&L8c16&@m40c@m0@L2"+w+"&L8d16&@m40d@m0ecf
cfgc@L2"+x+"&L8a16&@m40a@m0cb-<
 6420 z=bnd("f",6,6826+a,8192+a)
 6430 S(2)="@L2"+y+"&L8c16&@m40c@m0>b-ab-@L2<"+z+"&L8f16&@m40f@m
0d c>g<d>g<e-c@L2"+u+"&L8b-16&@m40b-@m0<
 6440 S(3)="@L2"+y+"&L8c16&@m20c&@m40c2.@m0@b"+str$(8192+a)
 6450 /*
 6460 s=bnd("b-",12,8192+a,6826+a)
 6470 U(0)="o3r1 r1 rb-a-ga-b-rfr<@L2"+w+"&L8d16&@m40d4>@m0@L2"+
u+"&L8b-16&@m20b-&
 6480 U(1)="@m40@L2"+s+"@m0L8r@b"+str$(8192+a)
 6490 t=bnd("e-",6,6826+a,8192+a)
 6500 U(2)="r1 r1 re-<f>f<e->e-<f@L2"+w+"&L8d16&@m40d@m0@L2"+t+"
&L8e-16&@m40e-@m0
 6510 s=bnd("c",12,8192+a,6826+a)
 6520 U(3)="@L2"+y+"&L8c16&@m20c&@m40c&@L4"+s+"@m0@b"+str$(8192+
a)
 6530 /*
 6540 V(0)="o3r1 r1 r8@L2"+u+"&L8b-16a-ga-b-r<@L2"+w+"&d16r8@L2"
+y+"&L8c16&@m40c4@m0
 6550 s=bnd("f",12,8192+a,6826+a)
 6560 V(1)="@L2"+z+"&L8f16&@m20f&@m40@L2"+s+"@m0L8r@b"+str$(8192
+a)
 6570 V(2)="r1 r1 r8@L2"+z+"&L8f16&@m40f4@m0e-rdr e-d>a-<@L2"+v+
"&L8a-16&@m20a-4&@m40a-4
 6580 endfunc
 6590 func solo2()
 6600 str s[256],t[256],u[256],v[256],w[256],x[256],y[256],z[256
]
 6610 u=bnd("a-",6,6826,8192):v=bnd("c",6,6826,8192):w=bnd("b-",
6,6826,8192)
 6620 R(0)="@d0 @93 o6 q8 v13 @u127@L2"+u+"&L8a-16&@m40a-&@m80a-
4@m0g-rfr g-a-re-rd-r@L2"+v+"&L8c16&@m60c@m0q7d-e->q6
 6630 x=bnd("g-",6,6826,8192)
 6640 R(1)="@L2"+w+"&L8b-16&@m60b-@m0a-r@L2"+u+"&L8a-16&@m60a-@m
0ga-@L2"+x+"&L8g-16&@m60g-@m0f
 6650 R(2)="@L2"+x+"&L8g-16&@m60g-@m0q7a-ga-<e->r@L2"+u+"&L8a-16
&@m60a-@m0g
 6660 y=bnd("d-",6,6826,8192):z=bnd("e-",6,6826,8192)
 6670 R(3)="g-fg-<@L2"+y+"&L8d-16&@m60d-@m0>@L2"+x+"&L8g-16&@m60
g-@m0f a-ga-<@L2"+z+"&L8e-16&@m60e-@m0>
 6680 s=bnd("r",18,8192,6826)
 6690 R(4)="@L2"+u+"&L8a-16&@m60a-@m0g g-fg-@d127'g-<d->'@m60r8@
L4"+s+"@m0@d0
 6700 /*
 6710 T(0)="@d0 @93 o5 q8 v13 @u127@L2"+u+"&L8a-16&@m40a-&@m80a-
4@m0<@L2"+u+"&L8a-16&@m40a-&@m80a-4@m0
 6720 T(1)="@L2"+x+"&L8g-16&@m60g-@m0q7fg-r@L2"+u+"&L8a-16&@m60a
-4
 6730 t=bnd("f",6,6826,8192)
 6740 T(2)="e-fg-@L2"+y+"&L8d-16&@m60d-@m0e-@L2"+t+"&L8f16&@m60f
@m0 cd-e->@L2"+u+"&L8a-16&@m40a-4&@m80a-4@m0
 6750 T(3)="<g-fe-a-r@L2"+u+"&L8a-16&@m60a-@m0e- g-fe-b-r@L2"+u+
"&L8a-16&@m60a-@m0e- g-fe-<d-r@L2"+v+"&L8c16&@m60c@m0>b-
 6760 s=bnd("a-",18,8192,6826)
 6770 T(4)="g-fe->@L2"+u+"&L8a-16&@m40a-&@L4"+s+"@m0
 6780 /*
 6790 U(0)="@b8192q8@L2<"+w+"&L8b-16&b-&@m20b-@m0@L2"+t+"&L8f16&
f2&@m20f2@m0<e-dr@L2"+v+"&L8c16&
 6800 U(1)="c1&@m20c2...r16@m0>
 6810 U(2)="@b8192@L2"+w+"&L8b-16&b-&@m20b-@m0@L2"+t+"&L8f16&f2&
@m20f2@m0<e-fr>@L2"+w+"&L8b-16&
 6820 U(3)="b-4.&@m20b-2<@m0@L2"+v+"&L8c16&
 6830 U(4)="c2&@m20c4..r16@m0>>
 6840 endfunc
```

THE SENTINEL

●リンカWLK

今月はリロケータブルファイルを実行可能なマシン語ファイルに変換するリンカWLKの登場です。マシン語プログラムを開発する場合には先月発表したWZDと今月のWLKの２つが必要ですので注意してください。WZDとWLKはいわば表裏一体で，どちらが欠けても用をなしません。

それならば，なぜまとめてひとつのプログラムにしてしまわないのかと疑問を持たれるかもしれませんね。リロケータブルファイルはマシン語ファイルの一種ですが，CALL命令やJP命令の飛び先などのアドレスがまだ確定していない不完全なマシン語ファイルです。

逆にリロケータブルファイルのこの特性を利用すると，CALL先がファイル内になくてもアセンブルできるというメリットが生まれます。これらの不完全なマシン語ファイルをいくつかつなぎあわせ，未確定のアドレスなどを確定して完全なマシン語ファイルにするのがリンカの役割です。

リンカの役割はリロケータブルファイルを実行可能なマシン語ファイルにすることだけであり，そのリロケータブルファイルがどんな処理系によって出力されようと知ったことではないのです。

第97部

リロケータブルファイルを出力するのはＣコンパイラかもしれないし，PASCALコンパイラかもしれません。JP先やCALL先がプログラム内になくてもいいのですから，メインとなる部分はＣ言語を使って書き，速度を要求される部分や緻密な処理を行いたい部分だけをアセンブリ言語で書くという技も使えます。リロケータブルファイルを扱う世界ではシステムプログラムはリロケータブルファイルを出力するものとそれを実行可能なマシン語ファイルに交換するものの２つに大きく分けることができるといえるでしょう。

WZDは前者に当たります。数ある（今後登場するかもしれない）リロケータブルファイルを出力するもののひとつにすぎません。そしてWLKは後者です。表裏一体となって使われる２つのプログラムがひとつにまとめられていない理由がおわかりいただけたでしょうか。

●四方山話

編集室に面白いゲームが届きました。編集者とライターの対戦が盛り上がっています。一世を風靡したこのゲームを来月はお届けすることにし，ライブラリアンは再来月ということになりました。ご期待ください。

```
# WLK.OBJ WLK1
-LBLOFF    3000 -NXTOFF   3000 -FLGCD    3000 -FLGDT    3000
-FLGEX     3000 -EXADR    3000 -PTRCD    3000 -PTRDT    3000
-PTRWK     3000 -CNTDT    3000 -wkout    3000 -SEGMD    3000
-iniwk1    301A -iniwk2   301D -PRTMAP   306A -RDOPEN   313B
-LBLBUF    315F -SEALBL   316C -FSEEK    3182 -asm      3193
-WROPEN    334C -PUTMAP   3355 -CLOSE    3358 -PRINT_   3594
-INPUT_    3724
  25 Undefined Label(s)
*
```

●S-OSの系譜（12）

S-OS“SWORD”はCP/MやMS-DOSなどのDOSと比較して足りない機能がいくつか存在していました。ひとつは何度か書いたようにファイルの扱いが弱いということです。ファイルとの１文字単位の入出力はサポートされていませんし，ファイル内をシークするなんてこともできません。しかし，こういった機能をほしい人（つまりファイル処理を行うS-OSのアプリケーションを自分で作成しようとする人）なら，用意されている機能を使って好きなように実現できる機能ですから，声高に叫ぶほどのことではないでしょう。

それより，掲載されたプログラムを入力して使う人にとってはユーザーフレンドリでないという点のほうが問題だったのではないでしょうか。CP/MやMS-DOSがユーザーフレンドリだと思う人はいないでしょうが,S-OSのユーザーインタフェイスはそれに輪を掛けて冷たいものでした。

そもそもユーザーインタフェイスなどと呼べるシェルは存在せず，モニタという名のコマンド解釈プログラムが用意されているだけだったのだから当然です。プログラムを実行するには，まずマシン語ファイルをロードし，実行開始番地へジャンプする。まさにマシン語モニタ感覚です。

1986年10月号では，このモニタを機能拡張しようという試みが行われました。まず，マシン語ファイルのロード・実行を一気に行うＲコマンドをモニタに追加。さらに，常用できる小さなプログラムとして，ファイルのコピープログラム(S-OSではASCIIファイルのコピーができない）とファイルの内容を画面に表示するプログラムが発売されました。これにより，モニタの使い勝手も随分向上しました。この試みは1987年５月号のS-OS“SWORD”変身セットへと受け継がれていくことになります。

同時に発表されたディスクモニタDREAMは，ディスク上のファイルを連続クラスタに収め直す，ディレクトリを並べ替えるといった便利な機能を備えたツールとして人気を集めました。

全機種共通S-OS"SWORD"要

リンカWLK

Ishigami Tatsuya
石上 達也

お待たせしました。ついにリンカWLKの登場です。これでWZDでアセンブルしたオブジェクトファイルを実際に起動できるファイルにすることができます。これらWZD，WLKはS-OSに新しい開発環境を築いてくれることでしょう。

WLKとは?

皆さん先月号のWZDは，もう入力し終わったでしょうか？　なにせ9Kバイトにも及ぶ超大作だったので大変だったでしょう。入力するのも大変ならデバッグするのは，その10倍大変なのです（システム関係は，ゲームと違いデバッグ＝気分転換にならない)。できるだけ多くの人に使っていただきたいものです。

さて，その9Kバイトにも及ぶWZDとともに大規模アプリケーションの開発に威力を発揮するのが，このプログラムです。ネーミングはWZD専用のリンカということで，WLKとしました。とくにアルファベットの語呂合せは，考えていません。今回は約7Kバイトの分量です。がんばって入力してください。

リンカとは?

先月号でも少しお話しましたが大規模なプログラム開発には，リロケータブルアセンブラを用いると有利です。そのとき，リロケータブルアセンブラから出力された個々のリロケータブルファイルをつなげ(リンクし) 1本のオブジェクトファイルを出力するのがこのリンカです。

ざっと使い方を説明しましょう。プロンプト'＊'が出ている状態が，入力可能状態です。ここでは，WZDのリンク作業を例にとって説明します。

まずWZDのメモリ配置は，3000H番地からコードセグメント，6000H番地からがデータセグメントですので，それぞれのスタートアドレスをリンカに知らせてやります。それには，

＊/P:3000,/D:6000

と入力します。そして，WZD本体は，WZD1,WZD2,WZD3,WZD35,WZD4から成っていますので，これらをリンクします。

＊WZD1

ここまで入力し終わったなら，画面上には，なにやら表が表示されるはずです。これは，WZD1をリンクした時点での，未定義ラベルとそのラベルが初めて使用された場所です。次に，

＊WZD2,WZD3,WZD35,WZD4

と，入力してください。今度はなにも表示されずに，プロンプト'＊'が表示されたはずです。なぜなら，WZDのすべてのファイルをリンクし終えたので，未定義のラベルはもうないからです。

この状態で，もしラベルリストがほしければ，

＊WZD/M

と入力してください。ファイル"WZD.MAP"にラベルリストが収録されています。

次に，WZDはデータセグメントに初期条件などを置いていないので(つまり，実際に必要なのはコードセグメントのみなので)，

＊WZD/N:P

と入力すると，"WZD.OBJ"というファイル名で，オブジェクトファイルを作ります。

＊WZD/N

と，最後にやってしまうとデータセグメントに割り振ったワークエリア領域までも含んだファイルができてしまいます。これでも一応は動くWZD.OBJができるのですが，かなりの無駄な部分まで含んでしまいます。

と，こんな具合にリンク作業は行うのですが，要は，/P：スイッチと/D：スイッチで，スタートアドレスを決めてやり，リンクしたいファイルの名前を打ち込んでやればいいのです。そうしたら，/Nスイッチで，それらをオブジェクトファイルとして，取り出せます。

プログラム

このプログラムもWZDと同様に腕力にものをいわせて作ったものです。それぞれのアイテムに応じて，適当な処理を行います。サブルーチンも何カ所か同じようなものが出てきたら，適宜作っていくという感じです（WZDと違いリンカは，これでも作れた)。

サブルーチンごとに，ほとんどが独立しているので解析はそんなに困難ではないと思います。

WZDと同様にWLK1.ASMは，最初Small-Cを用いて作成しました。このとき，

1)　ローカル変数は，だいたい1関数につき3つ以内（BC，DE，HLレジスタに対応させることができる)
2)　グローバルなポインタは2つ以内(IX，IYレジスタに対応させることができる)

という条件が満たされていたのでハンドコンパイルは，たいへんスムーズに行えました。

気になるサイズですが（プログラムのサイズです。念のため)，だいたいSmall-Cでコンパイルしたときの1/3程度になりました（ライブラリは除いて。本体のみ)。

メモリマップの変更は，リンカに与えるパラメータとWLKのヘッダファイルであるWLK.Hの内容を変更することによって行えます。現在，ローカルなラベルは1024個使用できるようになっていますが，この数を変更するときもこのWLK.Hを書き換えます。

WLKをソースからアセンブルするときは，

　# WZD
　*=WLK1
　*=WLK2
　*=WLK3
　*［ここでシフト+ブレイクを押す］
　# WLK
　*/P:3000,/D:4500
　*WLK1,WLK2,WLK3,WLK/N:P
とすれば，ここに掲載されているものと同様なオブジェクトが得られます。

いうまでもないと思いますが，ソースリストのみを一生懸命に入力しても，WZDとWLKのオブジェクト形式のプログラムがなければ，アセンブラを通してオブジェクトを得るということはできません。まず最初に，WZDとWLKのオブジェクト・プログラムが絶対に必要です。

最後にばらしてしまいますが，べき乗を行うサブルーチンは，第1パラメータ(HLレジスタ）を，第2パラメータ（DEレジスタ）回掛け合わせるということをしています。本当は，

　A^B=EXP(B*Log(A))

を展開して，ゴリゴリ計算したかったのですが，メモリを大量に使ううえ，浮動小数のレベルで計算しなければ精度が出ないので，しかたなしに，中学1年生しています。腕に自信のある方，なにかよい方法をご存じの方はご連絡ください。

WZDの訂正

コマンドラインからのRUNコマンドを拡張した“SWORD”を使用した場合，処理を終了して“SWORD”のモニタに戻ってくるときに誤動作する場合があります。

```
3008 ED 5B 76 IF → CD AB 50 00
303A 2A 76 IF    → CD B2 50
3153 ED 7B 00    → C3 FA IF
347A ED 7B 00    → C3 FA IF
```

以上のように訂正したうえ，次のダンプリストのようにプログラム（50B7Hまで）を追加してください。

```
5080 3E 09 37 18 04 3E 80 90 : E8
5088 A7 E1 C1 C9 26 00 6F 29 : D0
5090 29 29 29 C9 E5 CB 3C CB : FB
5098 1D CB 3C CB 1D CB 3C CB : DE
50A0 1D CB 3C CB 1D 7D E1 C9 : 33
50A8 00 00 00 ED 5B 76 1F 13 : F0
50B0 13 C9 2A 76 1F 23 23 C9 : AA
50B8 00 00 00 00 00 00 00 00 : 00
50C0 00 00 00 00 00 00 00 00 : 00
50C8 00 00 00 00 00 00 00 00 : 00
50D0 00 00 00 00 00 00 00 00 : 00
50D8 00 00 00 00 00 00 00 00 : 00
50E0 00 00 00 00 00 00 00 00 : 00
50E8 00 00 00 00 00 00 00 00 : 00
50F0 00 00 00 00 00 00 00 00 : 00
50F8 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 5B 72 C3 A3 C3 EA 8A F4 5C85
```

お詫びと訂正

実は，先月号の私の記事に少しばかり誤りがありましたので，ここに訂正させていただきます。まず，プログラムリストの呼び方が，記事中と注記中では，異なっていたこと。注記中のリスト2から6までは，それぞれ順に，

　WZD1,WZD2,WZD3,WZD35,WZD4

です。

また，79ページの中段31行目を，

　*WZD1,WZD2,WZD3,WZD35,WZD4,
　WZD/N:P

としてください。

＊　＊　＊

6月16日のコンサートにきてくださった皆様ありがとうございました。おかげさまで，コンサートは大成功をおさめることができました。ちなみに，コンサートのプログラムの役員紹介の写真で，コンサート委員長をひっくり返して下から支えているのが私です。来年もやりますので，ぜひ，またきてください。

表1

■起動方法

“SWORD”の拡張をしていない人はコマンドラインから，
　#LWLK
　#J3000
と，拡張をしてある人は，
　#WLK
で起動します。すると，この後ろにプロンプト‘*’を表示してパラメータの入力待ちになります。
なお，それぞれ，
　#J3000
　# WLK
の後ろにパラメータを書くことができます。

■パラメータ

以下，［ファイル名］とあるのは，大文字小文字を区別し，スイッチ（/Sとか/Nとか）は，どちらでもかまいません。

＊［ファイル名］
［ファイル名］で表されたリロケータブルファイルを，取り込みます。省略時の拡張子は‘.REL’です。

＊［ファイル名］/S
［ファイル名］で表されたライブラリファイルを，未定義なモジュールに限り取り込みます。省略時の拡張子は，‘.LIB’です。ライブラリファイルとかモジュールとかは，ライブラリアンWLBのときに詳しく説明します。

＊/P：xxxx
コードセグメントの領域を，アドレスxxxx（16進数で4桁以内）から取ります。

＊/D：xxxx
データセグメントの領域を，アドレスxxxx（16進数で4桁以内）から取ります。

＊［ファイル名］/N
リンクの結果を［ファイル名］で表されるファイルにコードセグメント，データセグメントともに出力します。デフォルトの拡張子は，‘.OBJ’です。以下同様。

＊［ファイル名］/N：P
リンクの結果を［ファイル名］で表されるファイルにコードセグメントのみ出力します。

＊［ファイル名］/N：D
リンクの結果を［ファイル名］で表されるファイルにデータセグメントのみ出力します。

＊［ファイル名］/M
ラベル情報を［ファイル名］で表されるファイルに出力する（外部ラベルの情報のみ）。

シフト+ブレイク
リンカの作業を中断し“SWORD”のコマンドラインに戻ります。

表2　エラーメッセージ

Undefined Item-xx
未定義アイテムxxを使用した。つまり指定されたリロケータブルファイルの内容がおかしい。WZDを使用している場合には起こらない。

Multi Defined
同じラベル名が，2カ所以上で定義されている。このメッセージの後ろに16進2桁の数字が表示されたらそのラベル番号を持つ内部ラベルがエラーであり，文字列が表示されたらその名前の外部ラベルがエラーである。

Undefined Label
未定義ラベルが使用された。内部エラーと外部エラーの区別については，Multi Definedエラーのときと同様である。

Too Many Labels
使用されたラベルの数が多すぎる。

Too Far
相対ジャンプ関係のオブジェクトを作成しようとしたが，目的のアドレスが相対ジャンプで届く範囲にない。

Stack Over
演算用のスタックがオーバーした（16レベル以上のスタックが使用された）。

Stack Empty
演算用のスタックが空なのにその内容を参照するアイテムが使用された。

Illegal ORG to xxxx
PC（Position Counter）を後ろ向きにアドレスxxxxに変更しようとした。以下のオブジェクトファイルの内容は保証されない。

DSEG buffer is Over flowed !!
DSEGエリア用のバッファが足りなくなった（DSEG用のバッファは8Kバイト，これ以上のメモリを用いる場合は，WLK.H内のメモリテーブルを変更するか，ほかのセグメントに割り振ってください）。

リスト1

```
3000 ED 7B 6A 1F ED 5B 76 1F : CE
3008 13 13 06 01 1A 13 A7 28 : 29
3010 07 FE 20 20 F7 04 18 F4 : 4C
3018 C5 CD 23 40 CD 45 40 21 : 68
3020 00 30 22 46 45 22 44 45 : 88
3028 22 42 45 22 58 45 22 56 : E0
3030 45 AF 32 3D 45 32 14 45 : 33
3038 32 28 45 32 3C 45 32 48 : CC
3040 45 32 49 45 DD 21 00 50 : 53
3048 F1 FE 01 20 05 CD 0B 34 : 21
3050 18 11 FD 2A 76 1F FD 23 : 05
3058 FD 23 FD 7E 00 FD 23 FE : B9
3060 20 20 F7 FD 7E 00 A7 20 : 79
3068 06 CD 92 3C CD 0B 34 11 : BE
3070 00 45 FD E5 E1 CD C6 36 : D1
3078 E5 FD E1 21 74 33 CD 77 : CF
-----------------------------------
SUM: BB 35 3C A3 E1 AA BA 07 5EC0

3080 36 21 78 33 C4 77 36 20 : 93
3088 34 CD EA 36 E5 ED 5B 56 : A4
3090 45 B7 ED 52 E1 DC 53 36 : 81
3098 22 56 45 DD 36 00 01 DD : AE
30A0 23 DD 75 00 DD 23 DD 74 : C6
30A8 00 DD 23 3A 48 45 A7 C2 : 30
30B0 B7 32 3E 01 32 48 45 22 : 09
30B8 42 45 C3 B7 32 21 7C 33 : 03
30C0 CD 77 36 21 80 33 C4 77 : 89
30C8 36 20 3A CD EA 36 E5 ED : 4F
30D0 5B 58 45 B7 ED 52 E1 DC : AB
30D8 53 36 22 58 45 22 5A 45 : 09
30E0 DD 36 00 02 DD 23 DD 75 : 67
30E8 00 DD 23 DD 74 00 DD 23 : 51
30F0 3A 49 45 A7 C2 B7 32 3E : 58
30F8 01 32 49 45 22 44 45 22 : 8E
-----------------------------------
SUM: B6 DF B5 52 1A 0C 3F 91 AD97

3100 46 45 C3 B7 32 21 84 33 : 0F
3108 CD 77 36 21 87 33 C4 77 : 90
3110 36 C2 A7 31 DD 36 00 04 : E7
3118 DD 23 21 00 45 CD 89 36 : F2
3120 11 00 45 21 D3 33 DC 93 : EC
3128 36 DD E5 D1 21 00 45 CD : FC
3130 AF 36 E5 DD E1 3E 01 11 : D8
3138 00 45 CD BC 40 DA 19 36 : 37
3140 CD 26 34 DD 22 3E 45 2A : D3
3148 3E 45 7E 23 FE FF CA 98 : 83
3150 31 FE E0 C2 34 36 5E 23 : BC
3158 56 23 ED 53 40 45 11 8D : DC
3160 45 7E 23 12 13 A7 20 F9 : CB
3168 22 3E 45 CD 14 3C D2 47 : DB
3170 31 CA 47 31 2A 40 45 DD : FF
3178 75 00 DD 23 DD 74 00 DD : A3
-----------------------------------
SUM: BB 0B A8 DC B2 F1 C1 F7 9559

3180 23 CD 82 42 DA 34 36 2A : 22
3188 B0 45 22 AE 45 3E 02 32 : 7C
3190 53 45 CD 28 37 C3 47 31 : FF
3198 DD 36 00 00 DD 23 DD 36 : 26
31A0 00 00 DD 23 C3 B7 32 21 : CD
31A8 8A 33 CD 77 36 21 8D 33 : 18
31B0 C4 77 36 20 1B 11 28 45 : 2A
31B8 21 00 45 CD AF 36 21 28 : 61
31C0 45 CD 89 36 11 28 45 21 : 70
31C8 D7 33 DC 93 36 C3 B7 32 : 5B
31D0 21 90 33 CD 77 36 21 95 : 14
31D8 33 C4 77 36 20 2B DD 36 : 02
31E0 00 FF 11 14 45 21 00 45 : CF
31E8 CD AF 36 21 14 45 CD 89 : 82
31F0 36 11 14 45 21 DB 33 DC : AB
31F8 93 36 3E 01 32 3C 45 32 : ED
-----------------------------------
SUM: 78 80 3E E6 80 40 A3 7E EB0A

3200 60 45 AF 32 61 45 C3 B7 : A6
3208 32 21 9A 33 CD 77 36 21 : BB
3210 9D 33 C4 77 36 20 2B DD : 69
3218 36 00 FF 11 14 45 21 00 : C0
3220 45 CD AF 36 21 14 45 CD : 3E
3228 89 36 11 14 45 21 DB 33 : 58
3230 DC 93 36 3E 01 32 3C 45 : 97
3238 32 61 45 AF 32 60 45 C3 : 21
3240 B7 32 21 A4 33 CD 77 36 : 5B
3248 21 A7 33 C4 77 36 20 2A : B6
3250 DD 36 00 FF 11 14 45 21 : 9D
3258 00 45 CD AF 36 21 14 45 : 71
3260 CD 89 36 11 14 45 21 DB : F2
3268 33 DC 93 36 3E 01 32 3C : 85
3270 45 32 61 45 32 60 45 C3 : B7
3278 B7 32 DD 36 00 03 DD 23 : FF
-----------------------------------
SUM: F2 AD 6F FC 86 C9 4B 80 36A3

3280 21 00 45 CD 89 36 11 00 : 03
3288 45 21 DF 33 DC 93 36 DD : FA
3290 E5 D1 21 00 45 CD AF 36 : CE
3298 E5 DD E1 3E 01 11 00 45 : 38
32A0 CD BC 40 21 00 45 DA FB : 04
32A8 35 2A B0 45 22 AE 45 3E : A7
32B0 02 32 53 45 CD 28 37 DD : D5
32B8 E5 E1 11 00 5F B7 ED 52 : 2C
32C0 38 0F 3A 3C 45 A7 20 09 : D2
32C8 21 AA 33 CD 17 37 C3 6D : 49
32D0 36 FD 7E 00 A7 21 CB 33 : 77
32D8 C4 77 36 28 0E 21 CD 33 : C8
32E0 CD 17 37 FD 2A 76 1F FD : D4
32E8 36 00 00 3A 3C 45 A7 CA : 62
32F0 63 30 CD 4D 34 ED 5B 42 : 6B
32F8 45 2A 56 45 B7 ED 52 21 : 21
-----------------------------------
SUM: 17 66 F5 E3 5B 2E 27 C6 F212

3300 E3 33 ED 4B 42 45 ED 5B : 1D
3308 56 45 1B C4 5D 33 ED 5B : 52
3310 44 45 2A 58 45 B7 ED 52 : 46
3318 21 EF 33 ED 4B 44 45 ED : F1
3320 5B 58 45 1B C4 5D 33 ED : 54
3328 5B 46 45 2A 5A 45 B7 ED : 53
3330 52 21 FB 33 ED 4B 46 45 : 64
3338 ED 5B 5A 45 1B C4 5D 33 : 56
3340 3A 28 45 A7 28 14 3E 04 : CC
3348 11 28 45 CD A6 41 21 28 : 7B
3350 45 DA FB 35 CD 28 3D CD : 4E
3358 FC 41 C3 FA 1F CD 17 37 : 34
3360 60 69 CD BE 1F 21 07 34 : CF
3368 CD 17 37 62 6B CD BE 1F : 92
3370 CD EE 1F C9 2F 50 3A 00 : 5C
3378 2F 70 3A 00 2F 44 3A 00 : 86
-----------------------------------
SUM: 48 0F E9 9D F7 F0 85 CA 0045

3380 2F 64 3A 00 2F 53 00 2F : 7E
3388 73 00 2F 4D 00 2F 6D 00 : 8B
3390 2F 4E 3A 50 00 2F 6E 3A : DE
3398 70 00 2F 4E 3A 44 00 2F : 9A
33A0 6E 3A 64 00 2F 4E 00 2F : B8
33A8 6E 00 53 6F 72 72 79 20 : AD
33B0 21 20 63 6F 6D 6D 61 6E : BC
33B8 64 73 20 61 72 65 20 74 : C3
33C0 6F 6F 20 6D 61 6E 79 20 : D3
33C8 21 0D 00 2C 00 3F 45 72 : 50
33D0 72 0D 00 4C 49 42 00 4D : A3
33D8 41 50 00 4F 42 4A 00 52 : BE
33E0 45 4C 00 43 53 45 47 20 : D3
33E8 61 72 65 61 20 3A 00 44 : 37
33F0 53 45 47 20 61 72 65 61 : 98
33F8 20 3A 00 57 53 45 47 20 : B0
-----------------------------------
SUM: FE 95 D8 79 FC F6 86 DF 0AB1

3400 61 72 65 61 20 3A 00 20 : 13
3408 2D 20 00 3E 2A CD F4 1F : 95
3410 ED 5B 76 1F CD D3 1F 1A : B6
3418 FE 1B CA 6D 36 13 1A A7 : 5A
3420 28 E9 D5 FD E1 C9 DD E5 : 4F
3428 E1 CD 22 37 36 FF FE FF : 39
3430 C8 FE E0 C2 34 36 36 E0 : E8
3438 23 CD 22 37 77 23 CD 22 : D2
3440 37 77 23 CD 22 37 77 23 : 91
3448 A7 20 F8 18 DC 21 BC 35 : C5
3450 CD 17 37 3E 01 32 3D 45 : 0E
3458 CD 45 40 2A 42 45 22 56 : 7B
3460 45 2A 44 45 22 58 45 22 : D9
3468 5A 45 3E 01 11 14 45 CD : 15
3470 A6 41 21 14 45 DA FB 35 : 6B
3478 DD 21 00 50 DD 7E 00 DD : 86
-----------------------------------
SUM: 07 4D D3 4F A5 A1 22 DA EFBE

3480 23 FE FF CA 52 35 FE 01 : 70
3488 20 21 DD 6E 00 DD 23 DD : 69
3490 66 00 DD 23 ED 5B 56 45 : 49
3498 22 56 45 B7 ED 52 DA 7C : 09
34A0 34 3A 60 45 A7 C4 EC 35 : 9F
34A8 C3 7C 34 FE 02 20 24 DD : 94
34B0 6E 00 DD 23 DD 66 00 DD : 8E
34B8 23 ED 5B 58 45 22 5A 45 : C9
34C0 22 58 45 B7 ED 52 DA 7C : 0B
34C8 34 3A 61 45 A7 C4 EC 35 : A0
34D0 C3 7C 34 FE 03 20 2E 21 : E3
34D8 C4 35 CD 17 37 DD E5 E1 : B7
34E0 CD 17 37 CD EE 1F 3E 01 : 34
34E8 DD E5 D1 CD BC 40 DD E5 : 1E
34F0 E1 DA FB 35 2A B0 45 22 : 2C
34F8 AE 45 3E 02 32 53 45 CD : CA
-----------------------------------
SUM: 69 76 B2 B2 CB A0 39 5B 23D0

3500 28 37 C3 7C 34 FE 04 C2 : 96
3508 7C 34 21 C4 35 CD 17 37 : E5
3510 DD E5 E1 CD 17 37 CD EE : 79
3518 1F 3E 01 DD E5 D1 CD BC : 7A
3520 40 DD E5 E1 DA FB 35 DD : CA
3528 7E 00 DD 23 A7 20 F8 DD : 1A
3530 6E 00 DD 23 DD 66 00 DD : 8E
3538 23 7C B5 CA 7C 34 CD 82 : 1D
3540 42 2A B0 45 22 AE 45 3E : B4
3548 02 32 53 45 CD 28 37 C3 : BB
3550 2F 35 3A 61 45 A7 CA 9D : 52
3558 35 2A 5E 45 7C B5 CA 9D : 9A
3560 35 ED 5B 56 45 2A 44 45 : CB
3568 37 ED 52 DA 7D 35 2A 44 : 70
3570 45 ED 5B 56 45 B7 ED 52 : 1E
3578 CD EC 35 18 0B 2A 5E 45 : DE
-----------------------------------
SUM: 15 55 F2 A9 01 FA 78 17 0A9D

3580 7C B5 21 CD 35 C4 17 37 : 66
3588 ED 4B 5E 45 21 00 90 7E : 0A
3590 23 C5 E5 CD C1 42 E1 C1 : 3F
3598 0B 78 B1 20 F2 2A 42 45 : F7
35A0 3A 60 45 A7 20 03 2A 44 : 17
35A8 45 22 00 46 3A 62 45 A7 : 35
35B0 28 03 2A 63 45 22 02 46 : 67
35B8 CD FC 41 C9 50 41 53 53 : 0A
35C0 2D 32 0D 00 4C 69 6E 6B : FA
35C8 69 6E 67 20 00 43 53 45 : 39
35D0 47 20 61 6E 64 20 44 53 : 51
35D8 45 47 20 61 72 65 20 70 : 74
35E0 69 6C 65 64 20 75 70 20 : C3
35E8 21 21 0D 00 7C B5 C8 E5 : 2D
35F0 AF CD C1 42 E1 DA 19 36 : 89
35F8 2B 18 F1 E5 21 0B 36 CD : 48
-----------------------------------
SUM: 91 37 DE 92 B8 38 3A BA EC7E

3600 17 37 E1 CD 17 37 CD EE : 05
3608 1F 18 62 43 61 6E 20 6E : 39
3610 6F 74 20 6F 70 65 6E 20 : D5
3618 00 21 21 36 CD 17 37 18 : AB
3620 4C 66 69 6C 65 20 41 63 : B0
3628 63 65 73 73 20 45 72 72 : F7
3630 6F 72 0D 00 21 3B 36 CD : 4D
3638 17 37 C9 49 6C 6C 65 67 : 04
3640 61 6C 20 4C 49 42 20 46 : 2A
3648 69 6C 65 20 45 72 72 6F : F2
3650 72 0D 00 21 5A 36 CD 17 : 14
3658 37 C9 49 6C 6C 65 67 61 : 4E
3660 6C 20 4F 52 47 20 45 72 : 4B
3668 72 6F 72 0D 00 3A 3D 45 : 1C
3670 A7 C4 FC 41 C3 FA 1F FD : 81
3678 E5 D1 7E A7 28 07 1A BE : E2
-----------------------------------
SUM: B7 2A 3F 1D 4D D7 61 3C B5CF

3680 C0 13 23 18 F5 D5 FD E1 : B6
3688 C9 7E 23 A7 37 C8 FE 2E : 3C
3690 20 F7 C9 06 0F 1A CD DE : BA
3698 36 38 09 05 28 06 FE 2E : D6
36A0 C8 13 18 F1 FE 2E C8 3E : 16
36A8 2E 12 13 CD AF 36 C9 06 : D4
36B0 14 AF 12 7E 23 CD DE 36 : 57
36B8 38 09 05 28 06 12 13 AF : 48
36C0 12 18 F0 EB 23 C9 06 14 : 0B
36C8 AF 12 7E CD DE 36 D8 05 : FD
36D0 C8 23 12 13 AF 12 18 F2 : DB
36D8 CD 17 37 C3 6D 36 FE 2F : AE
36E0 28 06 FE 2C 28 02 A7 C0 : E9
36E8 37 C9 21 00 00 FD 7E 00 : 9C
36F0 16 30 FE 30 D8 FE 3A 38 : BC
36F8 11 16 37 FE 41 D8 FE 47 : BA
-----------------------------------
SUM: FD 16 65 16 97 1C 99 BD 228E

3700 38 08 16 57 FE 61 D8 FE : E2
3708 67 D0 29 29 29 29 92 16 : 83
3710 00 5F 19 FD 23 18 D6 7E : 04
3718 A7 C8 23 E5 CD F4 1F E1 : 38
3720 18 F5 E5 CD 6E 42 E1 C9 : 19
3728 CD 6E 42 32 50 45 FE FF : 41
3730 C8 21 28 37 E5 ED 73 65 : F2
3738 45 E6 E0 CA 9D 38 FE 20 : C8
3740 CA C8 38 FE 60 CA 10 39 : 3B
3748 3A 50 45 E6 F8 FE 90 CA : 05
3750 EE 39 FE 98 CA 00 3A FE : BF
3758 A0 CA 12 3A FE A8 CA 2B : 51
3760 3A 3A 50 45 E6 7F 26 00 : 94
3768 6F 29 11 73 37 19 7E 23 : 0D
3770 66 6F E9 22 39 5F 39 89 : 3A
3778 39 97 39 A5 39 AD 39 DA : A7
-----------------------------------
SUM: 12 ED BA 97 06 56 69 72 6A72

3780 39 E7 39 E5 3E E5 3E E5 : 84
3788 3E E5 3E E5 3E E5 3E E5 : 8C
3790 3E E5 3E 73 38 73 38 73 : 2A
3798 38 73 38 73 38 73 38 73 : AC
37A0 38 73 38 73 38 73 38 73 : AC
37A8 38 73 38 73 38 73 38 73 : AC
37B0 38 73 38 73 38 73 38 73 : AC
37B8 38 73 38 73 38 73 38 73 : AC
37C0 38 73 38 73 38 73 38 73 : AC
37C8 38 73 38 73 38 73 38 73 : AC
37D0 38 73 38 2E 3A E5 3E E5 : 53
37D8 3E E5 3E E5 3E E5 3E E5 : 8C
37E0 3E E5 3E E5 3E E5 3E E5 : 8C
37E8 3E E5 3E E5 3E E5 3E E5 : 8C
37F0 3E E5 3E 3D 3A 48 3A 55 : AF
37F8 3A 62 3A 6F 3A 79 3A 82 : B4
-----------------------------------
SUM: A7 3F A7 EB A4 B7 A8 CD BFF0

3800 3A 90 3A 9E 3A B8 3A C9 : 97
3808 3A D7 3A E4 3A F4 3A 04 : 9B
3810 3B 14 3B E5 3E E5 3E E5 : B5
3818 3E E5 3E E5 3E E5 3E E5 : 8C
3820 3E E5 3E E5 3E E5 3E E5 : 8C
3828 3E E5 3E E5 3E E5 3E E5 : 8C
3830 3E E5 3E 73 38 1E 3B 32 : 97
3838 3B 3E 3B 43 3B 4B 3B 52 : 0A
3840 3B 60 3B 67 3B 81 3B E5 : 19
3848 3E E5 3E E5 3E E5 3E E5 : 8C
3850 3E E5 3E E5 3E E5 3E E5 : 8C
3858 3E E5 3E E5 3E E5 3E E5 : 8C
3860 3E E5 3E E5 3E E5 3E E5 : 8C
3868 3E 8D 3B 97 3B 9E 3B A5 : 56
3870 3B 73 38 11 85 38 CD E5 : 66
3878 1F 3A 50 45 CD C1 1F CD : 68
-----------------------------------
SUM: AD 7B D8 B4 9F 55 3C B5 AE0F

3880 EE 1F C3 FA 1F 49 6E 74 : 14
3888 65 72 6E 61 6C 20 45 72 : E9
3890 72 6F 72 20 49 54 45 4D : A2
3898 2D 4E 6F 3A 00 CD 8F 3E : BE
38A0 ED 5B AE 45 19 E5 ED 5B : 81
38A8 B0 45 B7 ED 52 E1 38 03 : 07
38B0 22 B0 45 E5 3A 50 45 E6 : B1
38B8 1F 47 CD 5D 3C E1 3A 3D : 24
38C0 45 A7 3E 01 CC AD 3B C9 : A8
38C8 3A 50 45 E6 1F 47 CD 5D : 45
38D0 3C CD 14 3C 30 18 F5 CD : 63
38D8 9A 3E F1 3E 00 CC AD 3B : BB
38E0 3E 01 32 52 45 3A 3D 45 : C4
38E8 A7 C4 2A 3F 18 1E E5 11 : 00
38F0 00 60 19 7E A7 E1 20 0C : AB
38F8 3E 01 32 52 45 3A 3D 45 : C4
-----------------------------------
SUM: 48 0D B8 EB 19 CC 54 C7 1FF0

3900 A7 C2 F5 3E 29 11 00 70 : 46
3908 19 5E 23 56 CD F1 3D C9 : B4
3910 3A 50 45 E6 1F 3C 47 C5 : 1C
```


▶Z80's Barは読んでもわからないのでメアリーさんのところだけ読んでいる。7月号は出番が少なかったですね。
小倉　修(18)埼玉県

```
3918 CD 6E 42 CD 37 3E C1 10 : 90
3920 F6 C9 CD 8F 3E CD 5B AE : 4F
3928 45 19 E5 ED 5B B0 45 B7 : 37
3930 ED 52 E1 38 03 22 B0 45 : 72
3938 E5 11 00 60 19 7E E1 A7 : 75
3940 20 10 3E 01 32 52 45 3A : 72
3948 3D 45 A7 28 05 CD F5 3E : 56
3950 18 09 11 00 70 29 19 7E : 62
3958 23 66 6F CD E9 3D C9 CD : 81
3960 8F 3E ED 5B AE 45 19 E5 : 06
3968 ED 5B B0 45 B7 ED 52 E1 : 14
3970 38 03 22 B0 45 E5 11 00 : 48
3978 60 19 36 01 E1 29 11 00 : CB
---------------------------------
SUM: 80 9C 8C A2 1C 7E 1F E8 AC0D

3980 70 19 CD 0D 3E 73 23 72 : A9
3988 C9 CD 19 3E E5 7D CD 37 : 53
3990 3E E1 7C CD 37 3E C9 CD : 73
3998 19 3E E5 7C CD 37 3E E1 : DB
39A0 7D CD 37 3E C9 CD 19 3E : AC
39A8 7D CD 37 3E C9 CD 9A 3E : 2D
39B0 44 4D CD 19 3E 3A 3D 45 : 71
39B8 A7 CA 37 3E 17 ED 37 38 : 84
39C0 0B 24 25 20 0F 7D B7 F2 : A9
39C8 37 3E 18 08 24 20 05 7D : 5B
39D0 B7 FA 37 3E CD 37 3E C3 : 2B
39D8 19 3F CD 9A 3E E5 CD 8F : 3E
39E0 3E D1 19 CD DE 3D C9 CD : A6
39E8 9A 3E CD E9 3D C9 3A 50 : 1E
39F0 45 E6 07 3C 47 C5 CD 6E : B5
39F8 42 CD 37 3E C1 10 F6 C9 : 14
---------------------------------
SUM: E6 13 1E 97 8F BA B6 65 E416

3A00 3A 50 45 E6 07 3C 47 C5 : 04
3A08 CD 8F 3E CD DE 3D C1 10 : 53
3A10 F6 C9 3A 50 45 E6 07 3C : B7
3A18 47 C5 CD 8F 3E E5 7C CD : D4
3A20 37 3E E1 7D CD 37 3E C1 : D6
3A28 10 EF C9 C3 EE 39 CD 19 : 98
3A30 3E 7C B5 C8 E5 AF CD 37 : CF
3A38 3E E1 2B 18 F4 CD 11 3E : 72
3A40 CD 1D 3E 19 CD E9 3D C9 : FD
3A48 CD 11 3E CD 1D 3E B7 ED : E8
3A50 52 CD E9 3D C9 CD 11 3E : 2A
3A58 CD 1D 3E CD 69 40 CD E9 : 54
3A60 3D C9 CD 11 3E CD 1D 3E : 4A
3A68 CD 7C 40 CD E9 3D C9 CD : 12
3A70 1D 3E 6C 26 00 CD E9 3D : E0
3A78 C9 CD 1D 3E 26 00 CD E9 : CD
---------------------------------
SUM: B0 5F 4D E4 65 3B E2 3B 779C

3A80 3D C9 CD 1D 3E 7C 2F 67 : 40
3A88 7D 2F 6F 23 CD E9 3D C9 : FA
3A90 CD 11 3E CD 1D 3E CD 7C : 8D
3A98 40 EB CD E9 3D C9 CD 1D : D1
3AA0 3E CD 11 3E 44 4D 21 01 : 0D
3AA8 00 78 B1 CA E9 3D C5 D5 : B3
3AB0 CD 69 40 D1 C1 0B 18 F1 : 1C
3AB8 CD 11 3E CD 1D 3E CB 3C : 4B
3AC0 CB 1D 1D 20 F9 CD E9 3D : 11
3AC8 C9 CD 11 3E CD 1D 3E 29 : 36
3AD0 1D 20 FC CD E9 3D C9 CD : C2
3AD8 1D 3E 7C 2F 67 7D 2F 6F : 88
3AE0 CD E9 3D C9 CD 11 3E CD : A5
3AE8 1D 3E 7C B2 67 7D B3 6F : 8F
3AF0 CD E9 3D C9 CD 11 3E CD : A5
3AF8 1D 3E 7C B2 67 7D B3 6F : 8F
---------------------------------
SUM: 41 49 9F EC EE FF D0 E6 14A7

3B00 CD E9 3D C9 CD 11 3E CD : A5
3B08 1D 3E 7C AA 67 7D AB 6F : 7F
3B10 CD E9 3D C9 CD 1D 3E 7C : 60
3B18 65 6F CD E9 3D C9 CD 6E : CB
3B20 42 06 02 A7 28 07 06 03 : 29
3B28 3D 28 02 06 04 78 32 53 : 6E
3B30 45 C9 CD 19 3E 22 54 45 : ED
3B38 3E 01 32 51 45 C9 AF 32 : B1
3B40 51 45 C9 CD 19 3E 7D CD : CD
3B48 37 3E C9 CD 19 3E CD DE : 0D
3B50 3D C9 CD 19 3E E5 7C CD : 58
3B58 37 3E E1 7D CD 37 3E C9 : DE
3B60 CD 8F 3E CD E9 3D C9 CD : 23
3B68 9A 3E EB CD 19 3E B7 ED : 8B
3B70 52 C8 DA 3F 3F E5 AF CD : D3
3B78 37 3E E1 2B 7C B5 20 F5 : C7
---------------------------------
SUM: 0A D4 EA 70 E7 8B 82 B0 05CD

3B80 C9 CD 1D 3E 22 63 45 3E : F9
3B88 01 32 62 45 C9 CD 6E 42 : 20
3B90 A7 C8 CD F4 1F 18 F6 CD : 2A
3B98 1D 3E CD E9 3F C9 CD 1D : 03
3BA0 3E CD BE 1F C9 CD 1D 3E : D9
3BA8 7D CD C1 1F C9 22 11 3C : 62
3BB0 32 13 3C CD 14 3C D2 1F : 8F
3BB8 3F 20 42 CD 6E 3C 26 00 : 3E
3BC0 6F 29 44 4D 60 69 CD AF : 6E
3BC8 40 78 B1 20 F7 2B 2B ED : C3
3BD0 4B 8B 45 CD 9D 40 2A 8B : 7A
3BD8 45 01 00 00 CD 9D 40 3A : 2A
3BE0 13 3C CD 98 40 ED 4B 11 : 3D
3BE8 3C CD 9D 40 11 8D 45 1A : E3
3BF0 13 F5 CD 98 40 F1 A7 20 : 65
3BF8 F6 22 8B 45 C9 2A 5B 3C : 72
---------------------------------
SUM: 51 1F 12 27 78 7E 90 EB 17BB

3C00 23 23 3A 13 3C CD 98 40 : 74
3C08 ED 4B 11 3C CD 9D 40 B7 : E6
3C10 C9 00 00 00 CD 6E 3C 26 : 66
3C18 00 6F 29 22 5B 3C 2A 5B : D6
3C20 3C CD AF 40 78 B1 37 C8 : 20
3C28 ED 43 5B 3C 21 05 00 09 : F6
3C30 11 8D 45 1A A7 28 0A 13 : E9
3C38 47 CD AA 40 B8 28 F4 18 : EA
3C40 DD CD 94 1F A7 20 D7 2A : 25
3C48 5B 3C 23 23 CD AA 40 CD : 61
3C50 AF 40 60 69 B7 C0 3E 01 : 6E
3C58 B7 37 C9 00 00 21 8D 45 : AA
3C60 C5 E5 CD 6E 42 E1 C1 77 : 40
3C68 23 10 F5 36 00 C9 E5 C5 : D1
3C70 21 8D 45 06 00 7E 23 A7 : 41
3C78 28 04 80 47 18 F7 78 C1 : 3B
---------------------------------
SUM: 29 4D D4 E3 AE E4 96 55 1054

3C80 E1 C9 21 00 00 01 01 02 : CF
3C88 AF CD 9A 1F ED A1 EA 89 : 36
3C90 3C C9 AF 32 67 45 21 00 : B3
3C98 00 22 68 45 21 00 02 ED : DF
3CA0 5B 8B 45 E5 B7 ED 52 E1 : E7
3CA8 30 14 23 23 CD AA 40 A7 : E8
3CB0 20 05 CD E6 3C 18 E8 23 : 37
3CB8 23 CD 21 3D 18 E1 CD EB : FF
3CC0 1F 2A 68 45 7C B5 C8 CD : BC
3CC8 E9 3F 21 D1 3C CD 17 37 : 71
3CD0 C9 20 55 6E 64 65 66 69 : 44
3CD8 6E 65 64 20 4C 61 62 65 : CB
3CE0 6C 28 73 29 0D 00 3E 2D : A8
3CE8 CD F4 1F CD AF 40 C5 06 : 67
3CF0 0A CD AA 40 A7 28 08 CD : 65
3CF8 F4 1F 10 F5 CD 21 3D 04 : 47
---------------------------------
SUM: 10 E8 B6 90 E5 48 44 E4 BAB2

3D00 CD F1 1F 10 FB E3 CD BE : 56
3D08 1F CD F1 1F 2A 68 45 23 : F6
3D10 22 68 45 3A 67 45 3C E6 : D7
3D18 03 32 67 45 CC EB 1F E1 : 98
3D20 C9 CD AA 40 A7 C8 18 F9 : 00
3D28 AF 32 67 45 21 00 02 E5 : 95
3D30 ED 5B 8B 45 B7 ED 52 E1 : EF
3D38 D2 73 3D 23 23 3E 2D CD : 00
3D40 D4 3D CD AA 40 A7 28 1A : B1
3D48 CD AF 40 C5 CD 82 3D E3 : F0
3D50 29 11 00 70 19 7E 23 66 : CA
3D58 6F CD B7 3D CD 9B 3D E1 : B6
3D60 18 CD 23 23 CD 82 3D 11 : C8
3D68 7D 3D CD AE 3D CD 9B 3D : 17
3D70 C3 2F 3D 3E 0D CD D4 3D : 58
3D78 AF CD D4 3D C9 2A 2A 2A : D4
---------------------------------
SUM: 88 F5 5A 03 CD F6 A1 2D A46D

3D80 2A 00 06 0A CD AA 40 A7 : 98
3D88 28 08 CD D4 3D 10 F5 CD : E0
3D90 21 3D 04 3E 20 CD D4 3D : 9E
3D98 10 F9 C9 3A 67 45 3C E6 : DA
3DA0 03 32 67 45 3E 20 C2 D4 : D5
3DA8 3D 3E 0D C3 D4 3D 1A A7 : 1D
3DB0 C8 13 CD D4 3D 18 F7 7C : 44
3DB8 CD BC 3D 7D F5 0F 0F 0F : 65
3DC0 0F CD C5 3D F1 CD CA 3D : A3
3DC8 18 0A E6 0F F6 30 FE 3A : 75
3DD0 D8 C6 07 C9 C5 D5 E5 CD : BA
3DD8 C1 42 E1 D1 C1 C9 E5 7D : A1
3DE0 CD 37 3E E1 7C CD 37 3E : E1
3DE8 C9 D5 54 5D CD F1 3D D1 : 1B
3DF0 C9 3A 6A 45 FE 11 D2 35 : C8
3DF8 3F E5 D5 26 00 6F 29 11 : C8
---------------------------------
SUM: B6 87 82 3E 89 29 28 B3 E215

3E00 6B 45 19 D1 73 23 72 E1 : 83
3E08 3C 32 6A 45 C9 AF 32 52 : 19
3E10 45 E5 CD 1D 3E 54 5D E1 : E4
3E18 C9 AF 32 52 45 3A 6A 45 : 2A
3E20 A7 CA 3A 3F 3D 32 6A 45 : 08
3E28 D5 26 00 6F 29 11 6B 45 : 54
3E30 19 7E 23 66 6F D1 C9 CD : F6
3E38 B5 3E 26 00 6F 3A 3D 45 : 44
3E40 A7 C8 3A 53 45 FE 03 28 : 6A
3E48 0C FE 04 C8 3A 60 45 A7 : 5C
3E50 7D C4 C1 42 C9 3A 61 45 : ED
3E58 A7 C8 E5 2A 5E 45 23 22 : 66
3E60 5E 45 7C 11 00 90 19 D1 : AA
3E68 FE 20 30 02 73 C9 21 77 : 24
3E70 3E CD 17 37 C3 6D 36 44 : 03
3E78 53 45 47 20 62 75 66 66 : A2
---------------------------------
SUM: C3 80 F3 8A 41 C6 E8 1D FFA3

3E80 65 72 20 69 73 20 6F 76 : D8
3E88 65 72 20 21 21 0D 00 CD : 13
3E90 6E 42 F5 CD 6E 42 67 F1 : 7A
3E98 6F C9 2A 54 45 3A 51 45 : CB
3EA0 A7 C0 2A 56 45 3A 53 45 : FE
3EA8 FE 02 C8 2A 58 45 FE 03 : 90
3EB0 C8 2A 5A 45 C9 F5 E5 3A : 6E
3EB8 53 45 FE 03 28 0D FE 04 : D0
3EC0 28 12 2A 56 45 23 22 56 : 9A
3EC8 45 18 10 2A 58 45 23 22 : 79
3ED0 58 45 18 07 2A 5A 45 23 : A8
3ED8 22 5A 45 2A 54 45 23 22 : C9
3EE0 54 45 E1 F1 C9 11 4B 3F : CF
3EE8 CD E5 1F 3A 50 45 CD C1 : 2E
3EF0 1F CD EE 1F C9 E5 11 5A : 12
3EF8 3F CD E5 1F E1 CD BE 1F : 9B
---------------------------------
SUM: CD AD 13 8D B3 39 EF 35 C56E

3F00 CD EE 1F C9 E5 11 6D 3F : 45
3F08 CD E5 1F E1 CD BE 1F CD : 29
3F10 EE 1F C9 11 84 3F C3 44 : B1
3F18 3F 11 94 3F C3 44 3F 11 : 7A
3F20 9C 3F CD E5 1F 11 8D 45 : 8F
3F28 18 1A 11 B1 3F CD E5 1F : 04
3F30 11 8D 45 18 0F 11 C2 3F : 1C
3F38 18 0A 11 D1 3F 18 05 11 : 71
3F40 DD 3F 18 00 CD E5 1F CD : D2
3F48 EE 1F C9 55 6E 64 65 66 : C8
3F50 69 6E 65 64 20 49 54 45 : A2
3F58 4D 00 55 6E 64 65 66 69 : A8
3F60 6E 65 64 20 4C 61 62 65 : CB
3F68 6C 2D 4E 6F 00 4D 75 6C : 84
3F70 74 69 20 44 65 66 69 6E : E3
3F78 65 64 20 4C 61 62 65 6C : C9
---------------------------------
SUM: D8 1E 5C BF 76 C6 AA A1 29F6

3F80 2D 4E 6F 00 54 6F 6F 20 : 3C
3F88 4D 61 6E 79 20 4C 61 62 : C4
3F90 65 6C 73 00 54 6F 6F 20 : 96
3F98 46 61 72 00 4D 75 6C 74 : BB
3FA0 69 20 44 65 66 69 6E 65 : D4
3FA8 64 20 4C 61 62 65 6C 2D : 91
3FB0 00 55 6E 64 65 66 69 6E : C9
3FB8 65 64 20 4C 61 62 65 6C : C9
3FC0 2D 00 53 74 61 63 6B 20 : 43
3FC8 4F 76 65 72 66 6C 6F 77 : 54
3FD0 00 53 74 61 63 6B 20 45 : 5B
3FD8 6D 70 74 79 00 49 6C 6C : EB
3FE0 65 67 61 6C 20 4F 52 47 : A1
3FE8 00 D5 F5 AF 32 4F 45 06 : 45
3FF0 05 11 4F 45 0E 0A CD 13 : A2
3FF8 40 C6 30 1B 12 10 F5 6B : D3
---------------------------------
SUM: EA C1 55 2A 3F 70 12 95 F539

4000 62 06 04 7E FE 30 20 05 : 3D
4008 36 20 23 18 F6 CD E5 1F : 58
4010 F1 E1 C9 C5 AF 06 10 29 : 4E
4018 17 2C 91 30 02 2D 81 10 : C4
4020 F6 C1 C9 AF 32 51 45 32 : 29
4028 52 45 32 62 45 21 00 02 : 93
4030 22 8B 45 01 00 10 21 00 : 24
4038 60 36 00 23 0B 78 B1 20 : 0D
4040 F8 CD 82 3C C9 AF 32 6A : 97
4048 45 32 DD 42 3D 32 95 42 : DC
4050 21 00 00 22 B0 45 22 AE : 08
4058 45 22 56 45 22 58 45 22 : E3
4060 5A 45 22 54 45 22 5E 45 : 1F
4068 C9 44 4D 21 00 00 3E 10 : C9
4070 29 CB 23 CB 12 30 01 09 : 2E
4078 3D 20 F5 C9 42 4B 54 5D : 59
---------------------------------
SUM: 96 8F FD AE 98 45 CC E8 D220

4080 3E 10 26 00 CB 23 CB 12 : 3F
4088 29 E5 B7 ED 42 E1 38 03 : 10
4090 ED 42 13 3D 20 EE EB C9 : 41
4098 CD 9A 1F 23 C9 F5 79 CD : AD
40A0 9A 1F 23 78 CD 9A 1F 23 : FD
40A8 F1 C9 CD 94 1F 23 C9 F5 : 1B
40B0 CD 94 1F 4F 23 CD 94 1F : 72
40B8 47 23 F1 C9 CD A3 1F D8 : 8B
40C0 CD 25 41 D8 2A 74 1F 11 : D9
40C8 23 46 01 20 00 ED B0 3A : 61
40D0 5D 1F 32 53 46 AF 32 B4 : DC
40D8 45 CD A9 43 D8 06 10 0E : FA
40E0 00 3A 41 46 11 43 46 12 : 6D
40E8 13 FE 7F 30 0F 2A 62 1F : 7A
40F0 85 6F 30 01 24 7E 05 28 : F4
40F8 28 0C 18 EB 0D F5 79 87 : 39
---------------------------------
SUM: 12 7A 34 61 6B 0A 39 A7 3C0E

4100 87 87 87 4F F1 D6 80 81 : AC
4108 32 59 46 AF 32 54 46 01 : 4D
4110 37 00 11 B5 45 21 23 46 : CC
4118 ED B0 3E FF 32 95 42 B7 : 9A
4120 C9 3E 07 37 C9 3A 5D 1F : C4
4128 CD 44 41 D8 CD 4F 41 D8 : 5F
4130 3E 08 37 C0 E5 ED 5B 74 : DE
4138 1F 01 20 00 ED B0 E1 7E : 3C
4140 CD 99 41 C9 FE 41 38 04 : EB
4148 FE 45 3F D0 3E 03 C9 0E : 6A
4150 10 ED 5B 60 1F ED 53 55 : 6C
4158 46 2A 64 1F 3E 01 CD 00 : FF
4160 20 D8 06 08 22 57 46 7E : 43
4168 FE FF 28 1A B7 28 0B D5 : FE
4170 ED 5B 74 1F CD 8A 41 D1 : 44
4178 28 0D D5 11 20 00 19 D1 : 25
---------------------------------
SUM: 24 4F 71 EB 61 41 D1 C4 5EAB

4180 10 E2 13 0D 20 CF 3E AF : EE
4188 B7 C9 C5 E5 06 10 13 23 : 76
4190 1A BE 20 02 10 F8 E1 C1 : A4
4198 C9 E5 E6 87 21 1F 29 BE : 42
41A0 E1 C8 3E 06 37 C9 CD A3 : 5D
41A8 1F D8 CD AF 1F D8 01 20 : 8B
41B0 00 11 23 46 2A 74 1F ED : 24
41B8 B0 3A 5D 1F 32 53 46 2A : 5B
41C0 E1 27 22 57 46 2A DF 27 : F7
41C8 22 55 46 CD D8 43 D8 32 : AF
41D0 41 46 32 43 46 3E 80 32 : 32
41D8 44 46 AF 32 59 46 32 B4 : F0
41E0 45 32 54 46 21 00 00 22 : 54
41E8 35 46 01 37 00 11 EC 45 : F5
41F0 21 23 46 ED B0 3E 00 32 : 97
41F8 DD 42 B7 C9 21 EC 45 11 : 02
---------------------------------
SUM: 5A 1E 04 61 B8 8A 28 14 B113

4200 23 46 01 37 00 ED B0 3A : 78
4208 53 46 32 5D 1F 2A 35 46 : EC
4210 2C 2D C4 F0 42 3A 59 46 : 28
4218 2A 35 46 2C 2D 20 01 3C : 5B
4220 67 22 35 46 3E 01 ED 5B : 8B
4228 55 46 2A 64 1F CD 00 20 : 35
4230 D8 21 23 46 ED 5B 57 46 : 47
4238 01 20 00 ED B0 3E 01 ED : EA
4240 5B 55 46 2A 64 1F CD 03 : 73
4248 20 D8 CD A9 43 D8 06 10 : 9F
4250 21 43 46 7E FE 7F 30 12 : E7
4258 23 4E E5 2A 62 1F 16 00 : 17
4260 5F 19 71 E1 05 CA 21 41 : FB
4268 18 E9 CD C7 43 C9 21 95 : 57
4270 42 34 20 08 CD 97 42 D8 : 1C
4278 21 E6 45 34 2A 95 42 7E : FF
---------------------------------
SUM: FA 71 A0 EC CE 2C 63 01 A6F9
```


▶この前，電器屋さんでMZ-2200の中古がデータレコーダとセットで8,000円で，売られていました。別のところでは初代PC-8001が本体のみ2,500円で売られてました。時の流れは速い。

本間　晃(19)愛知県

```
4280 B7 C9 7D 3D 32 95 42 7C : BF
4288 32 E6 45 CD 97 42 D8 21 : FC
4290 E6 45 34 B7 C9 00 B0 01 : 90
4298 37 00 11 23 46 21 B5 45 : CC
42A0 ED B0 3A 53 46 32 5D 1F : 1E
42A8 3A 54 46 47 3A 59 46 B8 : AC
42B0 DA 21 41 78 CD 8D 43 EB : 3C
42B8 3E 01 21 00 B0 CD 00 20 : FD
42C0 C9 2A FE 45 23 22 FE 45 : BE
42C8 2A DD 42 77 21 DD 42 34 : 34
42D0 37 3F C0 CD DF 42 D8 21 : 1D
42D8 1D 46 34 B7 C9 00 B1 01 : C9
42E0 37 00 11 23 46 21 EC 45 : 03
42E8 ED B0 3A 53 46 32 5D 1F : 1E
42F0 3A 54 46 47 3A 59 46 B8 : AC
42F8 30 76 CD A9 43 D8 3A 54 : C5
---------------------------------
SUM: 1A 20 7B 9C CA A2 F7 D0 5907

4300 46 E6 F0 47 3A 59 46 E6 : 22
4308 F0 B8 30 47 CB 3F CB 3F : 33
```

```
4310 CB 3F CB 3F 21 44 46 16 : D5
4318 00 5F 19 E5 3A 59 46 E6 : 1C
4320 F0 CD 8D 43 CD FA 43 2A : C1
4328 62 1F 16 00 5F 19 36 8F : D4
4330 CD D8 43 77 E1 D8 77 23 : B2
4338 36 80 2A 62 1F 16 00 5F : D6
4340 19 36 80 3A 59 46 E6 F0 : 7E
4348 C6 10 32 59 46 CD C7 43 : 7E
4350 D8 18 9D 3A 54 46 32 59 : EC
4358 46 CB 3F CB 3F CB 3F CB : 2F
4360 3F 21 44 46 16 00 5F 19 : 78
4368 3A 54 46 E6 0F C6 80 77 : 86
4370 3A 54 46 CD 8D 43 EB 3E : 9A
4378 01 21 00 B1 CD 03 20 D8 : 9B
---------------------------------
SUM: 07 93 72 10 3D 66 95 59 09A9

4380 01 37 00 11 EC 45 21 23 : BE
4388 46 ED B0 B7 C9 F5 F5 CB : 18
4390 3F CB 3F CB 3F CB 3F 21 : 7E
```

```
4398 43 46 16 00 5F 19 7E CD : 62
43A0 F2 43 F1 E6 0F 85 6F F1 : 00
43A8 C9 D5 E5 3A B4 45 57 3A : 47
43B0 5D 1F BA 28 0F 32 B4 45 : 98
43B8 3E 01 ED 5B 5E 1F 2A 62 : 90
43C0 1F CD 00 20 E1 D1 C9 D5 : 5C
43C8 E5 3E 01 ED 5B 5E 1F 2A : 13
43D0 62 1F CD 03 20 E1 D1 C9 : EC
43D8 C5 E5 06 80 2A 62 1F 7E : 59
43E0 B7 28 08 23 10 F9 3E 09 : 5A
43E8 37 18 04 3E 80 90 A7 E1 : 29
43F0 C1 C9 26 00 6F 29 29 29 : 9A
43F8 29 C9 E5 CB 3C CB 1D CB : 91
---------------------------------
SUM: 22 4E 6D F2 44 28 7A D2 ABAF

4400 3C CB 1D CB 3C CB 1D CB : DE
4408 3C CB 1D 7D E1 C9       : 4B
---------------------------------
SUM: 78 96 3A 48 1D 94 1D CB CF2A
```

リスト2

```
※000'                          ;==================================
0000'                          ;
0000'                          ; WLK Link Program Ver 1.00
0000'                          ;     Programed by T.Ishigami
0000'                          ;            '90 Mar 7th
0000'                          ;
0000'                          ;     CSEG 3000H -
0000'                          ;     DSEG 4500H -
0000'                          ;
0000'                          ;
0000'                          ;==================================
0000'
0000'                               CSEG
0000'
000F                           NAME EQU     15
0003                           EXT  EQU     3
0014                           LENFLEQU     NAME + 1 + EXT + 1
0000'
000D                           CR   EQU     0DH
001B                           BRK  EQU     1BH
FFFF                           EOF  EQU     -1
0000'
0001                           PRG  EQU     1
0002                           DTA  EQU     2
0003                           REL  EQU     3
0004                           LIB  EQU     4
0005                           MAP  EQU     5
0000'
0000'                        C      INCLUDE SOS.DEF
1FFA                         C _HOT EQU     1FFAH
1FF4                         C _PRINT       EQU 1FF4H
1FF1                         C _PRNTS       EQU 1FF1H
1FEE                         C _LTNL        EQU 1FEEH
1FEB                         C _NL  EQU     1FEBH
1FE5                         C _MSX EQU     1FE5H
1FDF                         C _TAB EQU     1FDFH
1FD3                         C _GETL        EQU 1FD3H
1FC7                         C _PAUSE       EQU 1FC7H
1FC1                         C _PRTHX       EQU 1FC1H
1FBE                         C _PRTHL       EQU 1FBEH
1FB8                         C _HEX EQU     1FB8H
1FA3                         C _FILE        EQU 1FA3H
1FAF                         C _WOPEN       EQU 1FAFH
1F9A                         C _POKE        EQU 1F9AH
1F94                         C _PEEK        EQU 1F94H
2009                         C _ROPEN       EQU 2009H
2015                         C _KILL        EQU 2015H
2012                         C _NAME        EQU 2012H
2033                         C _ERROR       EQU 2033H
0000'                        C
2000                         C _DRDSB       EQU 2000H
2003                         C _DWTSB       EQU 2003H
0000'                        C
1F7A                         C _PRCNT       EQU 1F7AH
1F76                         C _KBFAD       EQU 1F76H
1F74                         C _IBFAD       EQU 1F74H
1F72                         C _SIZE        EQU 1F72H
1F6A                         C _MEMAX       EQU 1F6AH
1F64                         C _DTBUF       EQU 1F64H
1F62                         C _FATBF       EQU 1F62H
1F60                         C _DIRPS       EQU 1F60H
1F5E                         C _FATPOS      EQU 1F5EH
1F5D                         C _DSK EQU     1F5DH
0000'                        C
0000'                        C      INCLUDE WLK.DEF
0000'                        C ; Header File For WLK
0000'                        C ;   CSEG 3000H-
0000'                        C ;   DSEG 4500H-
0000'                        C ;
1000                         C LBLMAX       EQU 1000H
0000'                        C
5000                         C cmdbuf       EQU 5000H    ;- 5FFFH
5F00                         C cmdlmt       EQU 5F00H
0000'                        C
6000                         C LBLFLG       EQU 6000H    ;- 6FFFH
7000                         C LBLNUM       EQU 7000H    ;- 8FFFH
0000'                        C
9000                         C BF_DSEG      EQU 9000H    ;- AFFFH
B000                         C RDBUF        EQU     0B000H   ;- B0FFH
B100                         C WRBUF        EQU     0B100H   ;- B1FFH
0000'
0000'                          ;
0000'                          ; Externals
0000'                          ;
                                    EXT     LBLOFF
                                    EXT     NXTOFF
                                    EXT     FLGCD
                                    EXT     FLGDT
                                    EXT     FLGEX
                                    EXT     EXADR
                                    EXT     PTRCD
                                    EXT     PTRDT
                                    EXT     PTRWK
                                    EXT     CNTDT
                                    EXT     wkout
                                    EXT     SEGMD
0000'                          ;
0000'                          ; Start
0000'                          main::
0000' ED 7B 6A 1F                   LD      SP,(_MEMAX)
0004'
0004' ED 5B 76 1F                   LD      DE,(_KBFAD)
0008' 13                            INC     DE
0009' 13                            INC     DE              ;Skip '# ' or'#J'
000A' 06 01                         LD      B,1             ;B = argc
000C' 1A                       CC1: LD      A,(DE)
000D' 13                            INC     DE
000E' A7                            AND     A
000F' 28 07                         JR      Z,CC3
0011' FE 20                         CP      ' '
0013' 20 F7                         JR      NZ,CC1
0015' 04                            INC     B
0016' 18 F4                         JR      CC1             ;if (DE) = ' ' then arg
c++
0018'
0018' C5                       CC3: PUSH    BC
0019'
0019' CD ** **                 CALL iniwk1
001C' CD ** **                 CALL iniwk2
001F' 21 00 30                 LD   HL,3000H
0022' 22 ** **                 LD   (strwk),HL
0025' 22 ** **                 LD   (strdt),HL
0028' 22 ** **                 LD   (strcd),HL
002B' 22 ** **                 LD   (PTRDT),HL
002E' 22 ** **                 LD   (PTRCD),HL
0031' AF                            XOR     A
0032' 32 ** **                 LD   (pass),A
0035' 32 ** **                 LD   (obname),A
0038' 32 ** **                 LD   (maname),A
003B' 32 ** **                 LD   (endflg),A
003E' 32 ** **                 LD   (sfgcd),A
0041' 32 ** **                 LD   (sfgdt),A
0044' DD 21 00 50                   LD      IX,cmdbuf       ;IX means Command p
ointer
0048'
0048' F1                            POP     AF              ;A = argc
0049' FE 01                         CP      1
004B' 20 05                         JR      NZ,CC6
004D' CD ** **                 CALL GETL
0050' 18 11                         JR      CC7
0052'
0052' FD 2A 76 1F              CC6: LD      IY,(_KBFAD)
0056' FD 23                         INC     IY
0058' FD 23                         INC     IY              ;Skip '# ' or '#J'
005A' FD 7E 00   CC4:          LD   A,(IY)
005D' FD 23                         INC     IY
005F' FE 20                         CP      ' '
0061' 20 F7                         JR      NZ,CC4
0063'
0063' FD 7E 00   CC7:          LD   A,(IY)
0066' A7                            AND     A
0067' 20 06                         JR      NZ,CC8
0069' CD ** **                 CALL PRTMAP
006C' CD ** **                 CALL GETL
006F'
006F'                          CC8:
006F' 11 ** **                 LD   DE,rename
0072' FD E5                         PUSH    IY
0074' E1                            POP     HL              ;LD HL,IY
0075' CD ** **                 CALL fcopy2
0078' E5                            PUSH    HL
0079' FD E1                         POP     IY              ;LD IY,HL
007B'
007B' 21 ** **                 LD   HL,swP
007E' CD ** **                 CALL amatch
0081' 21 ** **                 LD   HL,swP+4
0084' C4 ** **                 CALL NZ,amatch
0087' 20 34                         JR      NZ,CC9
0089'
0089' CD ** **                 CALL hlhex
008C' E5                            PUSH    HL
008D' ED 5B ** **                   LD      DE,(PTRCD)
0091' B7                            OR      A
0092' ED 52                         SBC     HL,DE
0094' E1                            POP     HL
0095' DC ** **                 CALL C,ERR14         ;Illegal ORG Error
0098'
0098' 22 ** **                 LD   (PTRCD),HL
009B'
009B' DD 36 00 01                   LD      (IX),PRG
009F' DD 23                         INC     IX
00A1' DD 75 00                 LD   (IX),L
00A4' DD 23                         INC     IX
00A6' DD 74 00                 LD   (IX),H
00A9' DD 23                         INC     IX
00AB'
00AB' 3A ** **                 LD   A,(sfgcd)
00AE' A7                            AND     A
00AF' C2 ** **                 JP   NZ,CC12
00B2'
00B2' 3E 01                         LD      A,1
00B4' 32 ** **                 LD   (sfgcd),A
00B7' 22 ** **                 LD   (strcd),HL
00BA' C3 ** **                 JP   CC12
00BD'
00BD' 21 ** **   CC9:          LD   HL,swD
00C0' CD ** **                 CALL amatch
00C3' 21 ** **                 LD   HL,swD+4
00C6' C4 ** **                 CALL NZ,amatch
00C9' 20 3A                         JR      NZ,CC13
00CB'
00CB' CD ** **                 CALL hlhex
00CE' E5                            PUSH    HL
00CF' ED 5B ** **                   LD      DE,(PTRDT)
00D3' B7                            OR      A
00D4' ED 52                         SBC     HL,DE
00D6' E1                            POP     HL
00D7' DC ** **                 CALL C,ERR14         ;Illegal ORG Error
00DA'
00DA' 22 ** **                 LD   (PTRDT),HL
00DD' 22 ** **                 LD   (PTRWK),HL
00E0'
00E0' DD 36 00 02                   LD      (IX),DTA
00E4' DD 23                         INC     IX
00E6' DD 75 00                 LD   (IX),L
00E9' DD 23                         INC     IX
00EB' DD 74 00                 LD   (IX),H
00EE' DD 23                         INC     IX
00F0'
00F0' 3A ** **                 LD   A,(sfgdt)
00F3' A7                            AND     A
00F4' C2 ** **                 JP   NZ,CC12
00F7' 3E 01                         LD      A,1
00F9' 32 ** **                 LD   (sfgdt),A
00FC' 22 ** **                 LD   (strdt),HL
00FF' 22 ** **                 LD   (strwk),HL
0102' C3 ** **                 JP   CC12
0105'
0105' 21 ** **   CC13:         LD   HL,swS
0108' CD ** **                 CALL amatch
010B' 21 ** **                 LD   HL,swS+3
010E' C4 ** **                 CALL NZ,amatch
0111' C2 ** **                 JP   NZ,CC17
0114'
0114' DD 36 00 04                   LD      (IX),LIB
0118' DD 23                         INC     IX
011A' 21 ** **                 LD   HL,rename
011D' CD ** **                 CALL there
0120'
0120' 11 ** **                 LD   DE,rename
0123' 21 ** **                 LD   HL,strLIB
0126' DC ** **                 CALL C,flneat
0129'
0129' DD E5                         PUSH    IX
012B' D1                            POP     DE      ;LD DE,IX
012C' 21 ** **                 LD   HL,rename
012F' CD ** **                 CALL fcopy
0132' E5                            PUSH    HL
0133' DD E1                         POP     IX      ;LD IX,HL
0135'
0135' 3E 01                         LD      A,1     ;Bin File
0137' 11 ** **                 LD   DE,rename
013A' CD ** **                 CALL ROOPEN
013D' DA ** **                 JP   C,Ferr
0140'
0140' CD ** **                 CALL gethead
0143' DD 22 ** **                   LD      (Hseek),IX
0147'
0147' 2A ** **   CC14:         LD   HL,(Hseek)
014A' 7E                            LD      A,(HL)
014B' 23                            INC     HL
014C'
014C' FE FF                         CP      EOF
014E' CA ** **                 JP   Z,CC16
0151' FE E0                         CP      0E0H
0153' C2 ** **                 JP   NZ,IFerr
0156'
0156' 5E                            LD      E,(HL)
0157' 23                            INC     HL
0158' 56                            LD      D,(HL)
0159' 23                            INC     HL
015A' ED 53 ** **                   LD      (Lseek),DE
015E'
015E' 11 ** **                 LD   DE,LBLBUF
0161' 7E                       CC15:LD      A,(HL)
0162' 23                            INC     HL
0163' 12                            LD      (DE),A
0164' 13                            INC     DE
0165' A7                            AND     A
0166' 20 F9                         JR      NZ,CC15
0168' 22 ** **                 LD   (Hseek),HL
016B'
016B' CD ** **                 CALL SEALBL
016E' D2 ** **                 JP   NC,CC14
0171' CA ** **                 JP   Z,CC14
0174'
0174' 2A ** **                 LD   HL,(Lseek)
0177' DD 75 00                 LD   (IX),L
017A' DD 23                         INC     IX
017C' DD 74 00                 LD   (IX),H
017F' DD 23                         INC     IX
0181'
0181' CD ** **                 CALL FSEEK
0184' DA ** **                 JP   C,IFerr
0187'
0187' 2A ** **                 LD   HL,(NXTOFF)
018A' 22 ** **                 LD   (LBLOFF),HL
018D' 3E 02                         LD      A,2     ;CSEG
018F' 32 ** **                 LD   (SEGMD),A
0192' CD ** **                 CALL asm
0195' C3 ** **                 JP   CC14
0198'
0198' DD 36 00 00              CC16:LD      (IX),0
019C' DD 23                         INC     IX
019E' DD 36 00 00                   LD      (IX),0
01A2' DD 23                         INC     IX
01A4' C3 ** **                 JP   CC12
01A7'
01A7' 21 ** **   CC17:         LD   HL,swM
01AA' CD ** **                 CALL amatch
01AD' 21 ** **                 LD   HL,swM+3
01B0' C4 ** **                 CALL NZ,amatch
01B3' 20 1B                         JR      NZ,CC22
01B5'
01B5' 11 ** **                 LD   DE,maname
01B8' 21 ** **                 LD   HL,rename
01BB' CD ** **                 CALL fcopy
01BE' 21 ** **                 LD   HL,maname
01C1' CD ** **                 CALL there
01C4'
01C4' 11 ** **                 LD   DE,maname
01C7' 21 ** **                 LD   HL,strMAP
01CA' DC ** **                 CALL C,flneat
01CD' C3 ** **                 JP   CC12
01D0'
01D0' 21 ** **   CC22:         LD   HL,swNP
01D3' CD ** **                 CALL amatch
01D6' 21 ** **                 LD   HL,swNP+5
01D9' C4 ** **                 CALL NZ,amatch
01DC' 20 2B                         JR      NZ,CC27
01DE'
01DE' DD 36 00 FF                   LD      (IX),EOF
01E2' 11 ** **                 LD   DE,obname
01E5' 21 ** **                 LD   HL,rename
01E8' CD ** **                 CALL fcopy
01EB' 21 ** **                 LD   HL,obname
01EE' CD ** **                 CALL there
01F1' 11 ** **                 LD   DE,obname
01F4' 21 ** **                 LD   HL,strOBJ
01F7' DC ** **                 CALL C,flneat
01FA'
01FA' 3E 01                         LD      A,1
01FC' 32 ** **                 LD   (endflg),A
01FF' 32 ** **                 LD   (FLGCD),A
0202' AF                            XOR     A
0203' 32 ** **                 LD   (FLGDT),A
0206' C3 ** **                 JP   CC12
0209'
0209' 21 ** **   CC27:         LD   HL,swND
020C' CD ** **                 CALL amatch
020F' 21 ** **                 LD   HL,swND+3
0212' C4 ** **                 CALL NZ,amatch
0215' 20 2B                         JR      NZ,CC32
0217'
0217' DD 36 00 FF                   LD      (IX),EOF
021B' 11 ** **                 LD   DE,obname
021E' 21 ** **                 LD   HL,rename
0221' CD ** **                 CALL fcopy
0224' 21 ** **                 LD   HL,obname
0227' CD ** **                 CALL there
022A' 11 ** **                 LD   DE,obname
022D' 21 ** **                 LD   HL,strOBJ
0230' DC ** **                 CALL C,flneat
```


▶東京書籍の公民（中３）の教科書の136ページと137ページの間の写真にあの「パピコン」が載っている。恐ろしいことよ……。
山野辺　太郎(14)宮城県

```
0233'
0233' 3E 01              LD     A,1
0235' 32 ** **      LD   (endflg),A
0238' 32 ** **      LD   (FLGDT),A
023B' AF                 XOR    A
023C' 32 ** **      LD   (FLGCD),A
023F' C3 ** **      JP   CC12
0242'
0242' 21 ** **  CC32:  LD   HL,swN
0245' CD ** **      CALL amatch
0248' 21 ** **      LD   HL,swN+3
024B' C4 ** **      CALL NZ,amatch
024E' 20 2A              JR     NZ,CC37
0250'
0250' DD 36 00 FF        LD     (IX),EOF
0254' 11 ** **      LD   DE,obname
0257' 21 ** **      LD   HL,rename
025A' CD ** **      CALL fcopy
025D' 21 ** **      LD   HL,obname
0260' CD ** **      CALL there
0263' 11 ** **      LD   DE,obname
0266' 21 ** **      LD   HL,strOBJ
0269' DC ** **      CALL C,flneat
026C'
026C' 3E 01              LD     A,1
026E' 32 ** **      LD   (endflg),A
0271' 32 ** **      LD   (FLGDT),A
0274' 32 ** **      LD   (FLGCD),A
0277' C3 ** **      JP   CC12
027A'
027A' DD 36 00 03   CC37:LD     (IX),REL
027E' DD 23              INC    IX
0280' 21 ** **      LD   HL,rename
0283' CD ** **      CALL there
0286' 11 ** **      LD   DE,rename
0289' 21 ** **      LD   HL,strREL
028C' DC ** **      CALL C,flneat
028F'
028F' DD E5              PUSH   IX
0291' D1                 POP    DE          ;LD DE,IX
0292' 21 ** **      LD   HL,rename
0295' CD ** **      CALL fcopy
0298' E5                 PUSH   HL
0299' DD E1              POP    IX          ;LD IX,HL
029B'
029B' 3E 01              LD     A,1         ;BIN file
029D' 11 ** **      LD   DE,rename
02A0' CD ** **      CALL RDOPEN
02A3' 21 ** **      LD   HL,rename
02A6' DA ** **      JP   C,openerr
02A9'
02A9' 2A ** **      LD   HL,(NXTOFF)
02AC' 22 ** **      LD   (LBLOFF),HL
02AF' 3E 02              LD     A,2     ;CSEG
02B1' 32 ** **      LD   (SEGMD),A
02B4'
02B4' CD ** **      CALL asm
02B7'
02B7' DD E5         CC12:PUSH   IX
02B9' E1                 POP    HL
02BA' 11 00 5F      LD   DE,cmdlmt
02BD' B7                 OR     A
02BE' ED 52              SBC    HL,DE
02C0' 38 0F              JR     C,CC44
02C2' 3A ** **      LD   A,(endflg)
02C5' A7                 AND    A
02C6' 20 09              JR     NZ,CC44
02C8'
02C8' 21 ** **      LD   HL,CC21
02CB' CD ** **      CALL _puts       ;puts("Command buffers over")
02CE' C3 ** **      JP   exit
02D1'
02D1' FD 7E 00  CC44:  LD   A,(IY)
02D4' A7                 AND    A
02D5'
02D5' 21 ** **      LD   HL,CC23
02D8' C4 ** **      CALL NZ,amatch
02DB' 28 0E              JR     Z,CC47
02DD'
02DD' 21 ** **  CC45:  LD   HL,CC23+2
02E0' CD ** **      CALL _puts       ;puts("?Err")
02E3' FD 2A 76 1F        LD     IY,(_KBFAD)
02E7' FD 36 00 00        LD     (IY),0
02EB'
02EB' 3A ** **  CC47:  LD   A,(endflg)
02EE' A7                 AND    A
02EF' CA ** **      JP   Z,CC7
02F2'
02F2' CD ** **      CALL pass2
02F5' ED 5B ** **        LD     DE,(strcd)
02F9' 2A ** **      LD   HL,(PTRCD)
02FC' B7                 OR     A
02FD' ED 52              SBC    HL,DE
02FF'
02FF' 21 ** **      LD   HL,CC50
0302' ED 4B ** **        LD     BC,(strcd)
0306' ED 5B ** **        LD     DE,(PTRCD)
030A' 1B                 DEC    DE
030B' C4 ** **      CALL NZ,prtadrs
030E'
030E' ED 5B ** **        LD     DE,(strdt)
0312' 2A ** **      LD   HL,(PTRDT)
0315' B7                 OR     A
0316' ED 52              SBC    HL,DE
0318'
0318' 21 ** **      LD   HL,CC51
031B' ED 4B ** **        LD     BC,(strdt)
031F' ED 5B ** **        LD     DE,(PTRDT)
0323' 1B                 DEC    DE
0324' C4 ** **      CALL NZ,prtadrs
0327'
0327' ED 5B ** **        LD     DE,(strwk)
032B' 2A ** **      LD   HL,(PTRWK)
032E' B7                 OR     A
032F' ED 52              SBC    HL,DE
0331'
0331' 21 ** **      LD   HL,CC52
0334' ED 4B ** **        LD     BC,(strwk)
0338' ED 5B ** **        LD     DE,(PTRWK)
033C' 1B                 DEC    DE
033D' C4 ** **      CALL NZ,prtadrs
0340'
0340' 3A ** **      LD   A,(maname)
0343' A7                 AND    A
0344' 28 14              JR     Z,CC48
0346'
0346' 3E 04              LD     A,4     ;ASC file
0348' 11 ** **      LD   DE,maname
034B' CD ** **      CALL WROPEN
034E' 21 ** **      LD   HL,maname
0351' DA ** **      JP   C,openerr
0354'
0354' CD ** **      CALL PUTMAP
0357'
0357' CD ** **      CALL CLOSE
035A'
035A' C3 FA 1F  CC48:  JP   _HOT
035D'
035D'
035D' CD ** **  prtadrs:CALL _puts
0360' 60                 LD     H,B
0361' 69                 LD     L,C
0362' CD BE 1F      CALL _PRTHL
0365' 21 ** **      LD   HL,CC53
0368' CD ** **      CALL _puts
036B' 62                 LD     H,D
036C' 6B                 LD     L,E
036D' CD BE 1F      CALL _PRTHL
0370' CD EE 1F      CALL _LTNL
0373' C9                 RET
0374'
0374' 2F 50 3A 00 2F  swP: DB     '/P:',0, '/p:',0
0379' 70 3A 00
037C' 2F 44 3A 00 2F  swD: DB     '/D:',0, '/d:',0
0381' 64 3A 00
0384' 2F 53 00 2F 73  swS: DB     '/S',0,  '/s',0
0389' 00
038A' 2F 4D 00 2F 6D  swM: DB     '/M',0,  '/m',0
038F' 00
0390' 2F 4E 3A 50 00  swNP:DB     '/N:P',0,'/n:p',0
0395' 2F 6E 3A 70 00
039A'
039A' 2F 4E 3A 44 00  swND:DB     '/N:D',0,'/n:d',0
039F' 2F 6E 3A 64 00
03A4'
03A4' 2F 4E 00 2F 6E  swN: DB     '/N',0,  '/n',0
03A9' 00
03AA'
03AA' 53 6F 72 72 79  CC21:DB     'Sorry ! commands are too many !',CR
,0
03AF' 20 21 20 63 6F
03B4' 6D 6D 61 6E 64
03B9' 73 20 61 72 65
03BE' 20 74 6F 6F 20
03C3' 6D 61 6E 79 20
03C8' 21 0D 00
03CB' 2C 00           CC23:DB     ',',0
03CD' 3F 45 72 72 0D       DB     '?Err',CR,0
03D2' 00
03D3'
03D3' 4C 49 42 00     strLIB:     DB 'LIB',0
03D7' 4D 41 50 00     strMAP:     DB 'MAP',0
03DB' 4F 42 4A 00     strOBJ:     DB 'OBJ',0
03DF' 52 45 4C 00     strREL:     DB 'REL',0
03E3'
03E3'
03E3' 43 53 45 47 20  CC50:DB     'CSEG area :',0
03E8' 61 72 65 61 20
03ED' 3A 00
03EF' 44 53 45 47 20  CC51:DB     'DSEG area :',0
03F4' 61 72 65 61 20
03F9' 3A 00
03FB' 57 53 45 47 20  CC52:DB     'WSEG area :',0
0400' 61 72 65 61 20
0405' 3A 00
0407' 20 2D 20 00     CC53:DB     ' - ',0
040B'
040B' 3E 2A           GETL:LD     A,'*'
040D' CD F4 1F        CALL _PRINT
0410' ED 5B 76 1F          LD     DE,(_KBFAD)
0414' CD D3 1F        CALL _GETL
0417' 1A                   LD     A,(DE)
0418' FE 1B                CP     BRK
041A' CA ** **        JP   Z,exit
041D' 13                   INC    DE          ;Skip Prompt '*'
041E' 1A                   LD     A,(DE)
041F' A7                   AND    A
0420' 28 E9                JR     Z,GETL         ;ﾓｼ CR ﾉﾐﾅﾗ ﾓｳｲﾁﾄﾞ
0422' D5                   PUSH   DE
0423' FD E1                POP    IY          ;LD IY,DE
0425' C9                   RET
0426'
0426'                 ; Get LIB File Header
0426' DD E5           gethead:PUSH IX
0428' E1                   POP    HL     ;LD HL,IX
0429' CD ** **  getl:  CALL INPUT
042C' 36 FF                LD     (HL),EOF
042E'
042E' FE FF                CP     EOF
0430' C8                   RET    Z
0431' FE E0                CP     0E0H
0433' C2 ** **        JP   NZ,IFerr
0436'
0436' 36 E0                LD     (HL),0E0H
0438' 23                   INC    HL
0439'
0439' CD ** **        CALL INPUT
043C' 77                   LD     (HL),A
043D' 23                   INC    HL
043E' CD ** **        CALL INPUT
0441' 77                   LD     (HL),A
0442' 23                   INC    HL
0443'
0443' CD ** **  get2:  CALL INPUT
0446' 77                   LD     (HL),A
0447' 23                   INC    HL
0448' A7                   AND    A
0449' 20 FB                JR     NZ,get2
044B' 18 DC                JR     getl
044D'
044D'                 ;
044D'                 ; PASS - 2
044D'                 ;
044D' 21 ** **  pass2: LD   HL,CC55
0450' CD ** **        CALL _puts        ;puts("PASS-2")
0453'
0453' 3E 01                LD     A,1
0455' 32 ** **        LD   (pass),A
0458'
0458' CD ** **        CALL iniwk2
045B' 2A ** **        LD   HL,(strcd)
045E' 22 ** **        LD   (PTRCD),HL
0461' 2A ** **        LD   HL,(strdt)
0464' 22 ** **        LD   (PTRDT),HL
0467' 22 ** **        LD   (PTRWK),HL
046A'                                   ;LD  A,(FLGDT)
046A'                                   ;AND A
046A'                                   ;JP  Z,CC56
046A'
046A'                                   ;;fopen("TMP.$$$")
046A'                                   ;CC56:
046A'
046A' 3E 01                LD     A,1     ;BIN file
046C' 11 ** **        LD   DE,obname
046F' CD ** **        CALL WROPEN
0472' 21 ** **        LD   HL,obname
0475' DA ** **        JP   C,openerr
0478'
0478' DD 21 00 50          LD     IX,cmdbuf
047C'                 CC59:
047C' DD 7E 00        LD   A,(IX)
047F' DD 23                INC    IX
0481' FE FF                CP     EOF
0483' CA ** **        JP   Z,CC60
0486'
0486' FE 01                CP     PRG
0488' 20 21                JR     NZ,CC65
048A'
048A' DD 6E 00        LD   L,(IX)
048D' DD 23                INC    IX
048F' DD 66 00        LD   H,(IX)
0492' DD 23                INC    IX
0494' ED 5B ** **          LD     DE,(PTRCD)
0498' 22 ** **        LD   (PTRCD),HL
049B' B7                   OR     A
049C' ED 52                SBC    HL,DE
049E' DA ** **        JP   C,CC59       ;Illegal ORG Error
04A1' 3A ** **        LD   A,(FLGCD)
04A4' A7                   AND    A
04A5' C4 ** **        CALL NZ,putspc
04A8' C3 ** **        JP   CC59
04AB'
04AB' FE 02           CC65:CP     DTA
04AD' 20 24                JR     NZ,CC66
04AF'
04AF' DD 6E 00        LD   L,(IX)
04B2' DD 23                INC    IX
04B4' DD 66 00        LD   H,(IX)
04B7' DD 23                INC    IX
04B9' ED 5B ** **          LD     DE,(PTRDT)
04BD' 22 ** **        LD   (PTRWK),HL
04C0' 22 ** **        LD   (PTRDT),HL
04C3' B7                   OR     A
04C4' ED 52                SBC    HL,DE
04C6' DA ** **        JP   C,CC59       ;Illegal ORG Error
04C9' 3A ** **        LD   A,(FLGDT)
04CC' A7                   AND    A
04CD' C4 ** **        CALL NZ,putspc
04D0' C3 ** **        JP   CC59
04D3'
04D3' FE 03           CC66:CP     REL
04D5' 20 2E                JR     NZ,CC68
04D7'
04D7' 21 ** **        LD   HL,CC56     ;printf("Linking %s¥n",IX)
04DA' CD ** **        CALL _puts
04DD' DD E5                PUSH   IX
04DF' E1                   POP    HL      ;LD HL,IX
04E0' CD ** **        CALL _puts
04E3' CD EE 1F        CALL _LTNL
04E6'
04E6' 3E 01                LD     A,1     ;BIN file
04E8' DD E5                PUSH   IX
04EA' D1                   POP    DE      ;LD DE,IX
04EB' CD ** **        CALL RDOPEN
04EE' DD E5                PUSH   IX
04F0' E1                   POP    HL      ;LD HL,IX
04F1' DA ** **        JP   C,openerr
04F4'                                     ;It is not necessary to skip
 IX
04F4'                                     ;Because File-name has not r
ode from 0 to 4
04F4' 2A ** **        LD   HL,(NXTOFF)
04F7' 22 ** **        LD   (LBLOFF),HL
04FA' 3E 02                LD     A,2     ;CSEG
04FC' 32 ** **        LD   (SEGMD),A
04FF'
04FF' CD ** **        CALL asm
0502'                                     ;;close(REL file)
0502' C3 ** **        JP   CC59
0505'
0505' FE 04           CC68:CP     LIB
0507' C2 ** **        JP   NZ,CC59
050A'
050A' 21 ** **        LD   HL,CC56     ;printf("Linking %s¥n",IX)
050D' CD ** **        CALL _puts
0510' DD E5                PUSH   IX
0512' E1                   POP    HL      ;LD HL,IX
0513' CD ** **        CALL _puts
0516' CD EE 1F        CALL _LTNL
0519'
0519' 3E 01                LD     A,1     ;BIN file
051B' DD E5                PUSH   IX
051D' D1                   POP    DE      ;LD DE,IX
051E' CD ** **        CALL RDOPEN
0521' DD E5                PUSH   IX
0523' E1                   POP    HL      ;LD HL,IX
0524' DA ** **        JP   C,openerr
0527'
0527' DD 7E 00  CC69:  LD   A,(IX)
052A' DD 23                INC    IX
052C' A7                   AND    A
052D' 20 F8                JR     NZ,CC69
052F'
052F' DD 6E 00  CC70:  LD   L,(IX)
0532' DD 23                INC    IX
0534' DD 66 00        LD   H,(IX)
0537' DD 23                INC    IX
0539' 7C                   LD     A,H
053A' B5                   OR     L
053B' CA ** **        JP   Z,CC59
053E'
053E' CD ** **        CALL FSEEK
0541'
0541' 2A ** **        LD   HL,(NXTOFF)
0544' 22 ** **        LD   (LBLOFF),HL
0547' 3E 02                LD     A,2     ;CSEG
0549' 32 ** **        LD   (SEGMD),A
054C' CD ** **        CALL asm
054F' C3 ** **        JP   CC70
0552'
0552'
0552'
0552' 3A ** **  CC60:  LD   A,(FLGDT)
0555' A7                   AND    A
0556' CA ** **        JP   Z,CC73
0559'
0559' 2A ** **        LD   HL,(CNTDT)
055C' 7C                   LD     A,H
055D' B5                   OR     L            ;There is the case in w
hich
055E' CA ** **        JP   Z,CC73          ;  FLGDT = YES & (CNTDT) = 0
0561'
0561'                                          ;;fclose("TMP.$$$")
0561'                                          ;;fopen("TMP.$$$")
0561' ED 5B ** **          LD     DE,(PTRCD)
0565' 2A ** **        LD   HL,(strdt)
0568' 37                   SCF
0569' ED 52                SBC    HL,DE
056B' DA ** **        JP   C,CC75      ;if (PTRCD) >= (strdt)  JP CC75
056E'
056E' 2A ** **        LD   HL,(strdt)
0571' ED 5B ** **          LD     DE,(PTRCD)
0575' B7                   OR     A
0576' ED 52                SBC    HL,DE
0578' CD ** **        CALL putspc
057B' 18 0B                JR     CC79
057D'
057D' 2A ** **  CC75:  LD   HL,(CNTDT)
0580' 7C                   LD     A,H
0581' B5                   OR     L
0582' 21 ** **        LD   HL,CC57
0585' C4 ** **        CALL NZ,_puts       ;"CSEG and DSEG are piled up
 !!"
0588'                 CC79:
0588' ED 4B ** **          LD     BC,(CNTDT)
058C' 21 00 90        LD   HL,BF_DSEG
058F'                 CC81:
058F' 7E                   LD     A,(HL)
0590' 23                   INC    HL
0591'
0591' C5                   PUSH   BC
0592' E5                   PUSH   HL
0593' CD ** **        CALL PRINT_
0596' E1                   POP    HL
0597' C1                   POP    BC
0598'
0598' 0B                   DEC    BC
0599' 78                   LD     A,B
059A' B1                   OR     C
059B' 20 F2                JR     NZ,CC81
059D'
059D'                                     ;;fclose("TMP.$$$")
059D'                                     ;;unlink("TMP.$$$")
059D' 2A ** **  CC73:  LD   HL,(strcd)
05A0' 3A ** **        LD   A,(FLGCD)
05A3' A7                   AND    A
05A4' 20 03                JR     NZ,CC74
05A6' 2A ** **        LD   HL,(strdt)
05A9' 22 ** **  CC74:  LD   (wkout+14H),HL  ;Set Start Address
05AC' 3A ** **        LD   A,(FLGEX)
05AF' A7                   AND    A
05B0' 28 03                JR     Z,CC76
05B2' 2A ** **        LD   HL,(EXADR)
05B5' 22 ** **  CC76:  LD   (wkout+16H),HL ;Set exec Address
05B8' CD ** **        CALL CLOSE
05BB' C9                   RET
05BC'
05BC' 50 41 53 53 2D  CC55:DB     'PASS-2',CR,0
05C1' 32 0D 00
05C4' 4C 69 6E 6B 69  CC56:DB     'Linking ',0
05C9' 6E 67 20 00
05CD' 43 53 45 47 20  CC57:DB     'CSEG and DSEG are piled up !!',CR,0
05D2' 61 6E 64 20 44
05D7' 53 45 47 20 61
05DC' 72 65 20 70 69
05E1' 6C 65 64 20 75
05E6' 70 20 21 21 0D
05EB' 00
05EC'
05EC'                 ; Put Space into OBJ-File
05EC'
05EC' 7C              putspc:     LD  A,H
05ED' B5                   OR     L
05EE' C8                   RET    Z
05EF'
05EF' E5                   PUSH   HL
05F0' AF                   XOR    A
05F1' CD ** **        CALL PRINT_
05F4' E1                   POP    HL
05F5' DA ** **        JP   C,Ferr
05F8' 2B                   DEC    HL
05F9' 18 F1                JR     putspc
05FB'
05FB' E5              openerr:PUSH HL
05FC' 21 ** **        LD   HL,CC83
05FF' CD ** **        CALL _puts
0602' E1                   POP    HL
0603' CD ** **        CALL _puts
0606' CD EE 1F        CALL _LTNL
0609' 18 62                JR     exit
060B' 43 61 6E 20 6F  CC83:DB     'Can not open ',0
0610' 6F 74 20 6F 70
0615' 65 6E 20 00
0619'
0619'
0619' 21 ** **  Ferr:  LD   HL,CC84
061C' CD ** **        CALL _puts
061F' 18 4C                JR     exit
0621' 66 69 6C 65 20  CC84:DB     'file Access Error',CR,0
```

THE SENTINEL

▶研究室でEWSを使っていてUNIXについていろいろと勉強していると家のX68000に同じような環境を望んでしまう。　川尻　博光(21)岐阜県

```
0626' 41 63 63 65 73
062B' 73 20 45 72 72
0630' 6F 72 0D 00
0634'
0634' 21 ** **  IFerr:  LD   HL,CC85
0637' CD ** **          CALL _puts
063A' C9                     RET
063B' 49 6C 6C 65 67    CC85:DB        'Illegal LIB File Error',CR,0
0640' 61 6C 20 4C 49
0645' 42 20 46 69 6C
064A' 65 20 45 72 72
064F' 6F 72 0D 00
0653'
0653' 21 ** **  ERR14:  LD   HL,STR14
0656' CD ** **          CALL _puts
0659' C9                     RET
065A' 49 6C 6C 65 67    STR14:         DB  'Illegal ORG Error',CR,0
065F' 61 6C 20 4F 52
0664' 47 20 45 72 72
0669' 6F 72 0D 00
066D'
066D' 3A ** **  exit::  LD   A,(pass)
0670' A7                     AND      A
0671' C4 ** **          CALL NZ,CLOSE
0674' C3 FA 1F          JP   _HOT
0677'
0677' FD E5             amatch:        PUSH     IY
0679' D1                     POP      DE       ;LD DE,IY
067A' 7E                CC88:LD       A,(HL)
067B' A7                     AND      A
067C' 28 07                  JR       Z,CC89
067E' 1A                     LD       A,(DE)
067F' BE                     CP       (HL)
0680' C0                     RET      NZ       ;Not Matched
0681' 13                     INC      DE
0682' 23                     INC      HL
0683' 18 F5                  JR       CC88
0685'
0685' D5                CC89:PUSH     DE
0686' FD E1                  POP      IY       ;LD IY,DE
0688' C9                     RET               ;Matched
0689'
0689' 7E                there:         LD  A,(HL)
068A' 23                     INC      HL
068B' A7                     AND      A
068C' 37                     SCF
068D' C8                     RET      Z        ;RET With CY = 1
068E' FE 2E                  CP       '.'
0690' 20 F7                  JR       NZ,there
0692' C9                     RET               ;RET With CY = 0
0693'
0693' 06 0F             flncat:        LD  B,NAME       ;B = Counter
0695' 1A                CC97:LD       A,(DE)
0696' CD ** **          CALL isflchr
0699' 38 09                  JR       C,CC98
069B' 05                     DEC      B
069C' 28 06                  JR       Z,CC98
069E'
069E' FE 2E                  CP       '.'
06A0' C8                     RET      Z
06A1' 13                     INC      DE
06A2' 18 F1                  JR       CC97
06A4'
06A4' FE 2E             CC98:CP       '.'
06A6' C8                     RET      Z
06A7' 3E 2E                  LD       A,'.'
06A9' 12                     LD       (DE),A
06AA' 13                     INC      DE
06AB' CD ** **          CALL fcopy
06AE' C9                     RET
06AF'
06AF' 06 14             fcopy:         LD  B,LENFL
06B1'
06B1' AF                     XOR      A
06B2' 12                     LD       (DE),A
06B3'
06B3' 7E                CC104:         LD  A,(HL)
06B4' 23                     INC      HL
06B5' CD ** **          CALL isflchr
06B8' 38 09                  JR       C,CC105
06BA' 05                     DEC      B
06BB' 28 06                  JR       Z,CC105
06BD'
06BD' 12                     LD       (DE),A
06BE' 13                     INC      DE
06BF' AF                     XOR      A
06C0' 12                     LD       (DE),A
06C1' 18 F0                  JR       CC104
06C3'
06C3' EB                CC105:         EX  DE,HL
06C4' 23                     INC      HL
06C5' C9                     RET
06C6'
06C6' 06 14             fcopy2:        LD  B,LENFL
06C8' AF                     XOR      A
06C9' 12                     LD       (DE),A
06CA' 7E                CC109:         LD  A,(HL)
06CB' CD ** **          CALL isflchr
06CE' D8                     RET      C
06CF' 05                     DEC      B
06D0' C8                     RET      Z
06D1'
06D1' 23                     INC      HL
06D2' 12                     LD       (DE),A
06D3' 13                     INC      DE
06D4' AF                     XOR      A
06D5' 12                     LD       (DE),A
06D6' 18 F2                  JR       CC109
06D8'
06D8' CD ** **  error:  CALL _puts
06DB' C3 ** **          JP   exit
06DE'
06DE' FE 2F             isflchr:CP    '/'
06E0' 28 06                  JR       Z,isfl1
06E2' FE 2C                  CP       ','
06E4' 28 02                  JR       Z,isfl1
06E6' A7                     AND      A
06E7' C0                     RET      NZ       ;CY = 0
06E8' 37                isfl1:         SCF
06E9' C9                     RET               ;CY = 1
06EA'
06EA' 21 00 00  hlhex:  LD   HL,0
06ED'
06ED' FD 7E 00  CC118:  LD   A,(IY)
06F0' 16 30                  LD       D,'0'
06F2' FE 30                  CP       '0'
06F4' D8                     RET      C
06F5' FE 3A                  CP       '9' + 1
06F7' 38 11                  JR       C,CC119
06F9'
06F9' 16 37                  LD       D,'A' - 10
06FB' FE 41                  CP       'A'
06FD' D8                     RET      C
06FE' FE 47                  CP       'F' + 1
0700' 38 08                  JR       C,CC119
0702'
0702' 16 57                  LD       D,'a' - 10
0704' FE 61                  CP       'a'
0706' D8                     RET      C
0707' FE 67                  CP       'f' + 1
0709' D0                     RET      NC
070A'
070A' 29                CC119:         ADD HL,HL
070B' 29                     ADD      HL,HL
070C' 29                     ADD      HL,HL
070D' 29                     ADD      HL,HL          ;HL = HL * 16
070E' 92                     SUB      D
070F' 16 00                  LD       D,0
0711' 5F                     LD       E,A
0712' 19                     ADD      HL,DE
0713' FD 23                  INC      IY
0715' 18 D6                  JR       CC118
0717'
0717'
0717' 7E                _puts::        LD  A,(HL)
0718' A7                     AND      A
0719' C8                     RET      Z
071A' 23                     INC      HL
071B' E5                     PUSH     HL
071C' CD F4 1F          CALL _PRINT
071F' E1                     POP      HL
0720' 18 F5                  JR       _puts
0722'
0722' E5                INPUT:         PUSH    HL
0723' CD ** **          CALL INPUT_
0726' E1                     POP      HL
0727' C9                     RET
0728'
0728'                   ;
0728'                   ; Works
0728'                   ;
0728'
0728'                        DSEG
0000"
0000"                   rename::DS    LENFL             ;Relocatble File Na
me
0014"                   obname::DS    LENFL             ;Object File name
0028"                   maname::DS    LENFL             ;Map File name
003C"
003C" 00                endflg::DB    0
003D" 00                pass::        DB  0
003E"                   Hseek:        DS  2
0040"                   Lseek:        DS  2             ;Seek Address
0042" 00 00             strcd:        DW  0             ;Start Address of C
SEG
0044" 00 00             strdt:        DW  0             ;Start Address of D
SEG
0046" 00 00             strwk:        DW  0             ;Start Address of W
SEG
0048" 00                sfgcd:        DB  0             ;Set Flag for CSEG
0049" 00                sfgdt:        DB  0             ;Set Flag for DSEG
004A"
004A"                   ;
004A"                   ; Externals
004A"                   ;
                             EXT      PRTMAP
                             EXT      asm
                             EXT      iniwk1
                             EXT      iniwk2
                             EXT      PUTMAP
                             EXT      SEALBL
                             EXT      LBLBUF
                             EXT      RDOPEN
                             EXT      WROPEN
                             EXT      CLOSE
                             EXT      FSEEK
                             EXT      PRINT_
                             EXT      INPUT_
004A"
004A"                        END
```

リスト3

```
0000'                      ;==================================
0000'                      ;
0000'                      ;  Linker For W-ZEDA
0000'                      ;
0000'                      ;    Programed By T.Ishigami
0000'                      ;             '90 Feb.22nd
0000'                      ;
0000'                      ;==================================
0000'
                                EXT     pass
                                EXT     _puts
                                EXT     exit
                                EXT     INPUT_
                                EXT     PRINT_
                                EXT     RDPNT
                                EXT     WRPNT
0000'
0000'                     C     INCLUDE SOS.DEF
1FFA                      C _HOT EQU    1FFAH
1FF4                      C _PRINT      EQU 1FF4H
1FF1                      C _PRNTS      EQU 1FF1H
1FEE                      C _LTNL       EQU 1FEEH
1FEB                      C _NL  EQU    1FEBH
1FE5                      C _MSX EQU    1FE5H
1FDF                      C _TAB EQU    1FDFH
1FD3                      C _GETL       EQU 1FD3H
1FC7                      C _PAUSE      EQU 1FC7H
1FC1                      C _PRTHX      EQU 1FC1H
1FBE                      C _PRTHL      EQU 1FBEH
1FB8                      C _HEX EQU    1FB8H
1FA3                      C _FILE       EQU 1FA3H
1FAF                      C _WOPEN      EQU 1FAFH
1F9A                      C _POKE       EQU 1F9AH
1F94                      C _PEEK       EQU 1F94H
2009                      C _ROPEN      EQU 2009H
2015                      C _KILL       EQU 2015H
2012                      C _NAME       EQU 2012H
2033                      C _ERROR      EQU 2033H
0000'                     C
2000                      C _DRDSB      EQU 2000H
2003                      C _DWTSB      EQU 2003H
0000'                     C
1F7A                      C _PRCNT      EQU 1F7AH
1F76                      C _KBFAD      EQU 1F76H
1F74                      C _IBFAD      EQU 1F74H
1F72                      C _SIZE       EQU 1F72H
1F6A                      C _MEMAX      EQU 1F6AH
1F64                      C _DTBUF      EQU 1F64H
1F62                      C _FATBF      EQU 1F62H
1F60                      C _DIRPS      EQU 1F60H
1F5E                      C _FATPOS     EQU 1F5EH
1F5D                      C _DSK EQU    1F5DH
0000'                     C
0000'                     C     INCLUDE WLK.DEF
0000'                     C ; Header File For WLK
0000'                     C ;   CSEG 3000H-
0000'                     C ;   DSEG 4500H-
0000'                     C ;
1000                      C LBLMAX      EQU 1000H
0000'                     C
5000                      C cmdbuf      EQU 5000H    ;- 5FFFH
5F00                      C cmdlmt      EQU 5F00H
0000'                     C
6000                      C LBLFLG  ,   EQU 6000H    ;- 6FFFH
7000                      C LBLNUM      EQU 7000H    ;- 8FFFH
0000'                     C
9000                      C BF_DSEG     EQU 9000H    ;- AFFFH
B000                      C RDBUF       EQU     0B000H   ;- B0FFH
B100                      C WRBUF       EQU     0B100H   ;- B1FFH
0000'
0000'
0010                        STMAXEQU    16      ;Stack Level Maxium
0000'
000D                        CR   EQU    0DH
0009                        TAB  EQU    09H
0000'
0000' CD ** **   asm::      CALL INPUT_
0003' 32 ** **              LD   (ITEM),A
0006' FE FF                      CP      0FFH
0008' C8                         RET     Z
0009'
0009' 21 ** **              LD   HL,asm
000C' E5                    .    PUSH    HL      ;Return Stack
000D' ED 73 ** **                LD      (SPBUF),SP
0011'                     ;      LD      A,(ITEM)
0011' E6 E0                      AND     0E0H
0013' CA ** **            JP     Z,ITM00
0016' FE 20                      CP      20H
0018' CA ** **            JP     Z,ITM20
001B' FE 60                      CP      60H
001D' CA ** **            JP     Z,ITM60
0020'
0020' 3A ** **            LD     A,(ITEM)
0023' E6 F8                      AND     0F8H
0025' FE 90                      CP      90H
0027' CA ** **            JP     Z,ITM90
002A' FE 98                      CP      98H
002C' CA ** **            JP     Z,ITM98
002F' FE A0                      CP      0A0H
0031' CA ** **            JP     Z,ITMA0
0034' FE A8                      CP      0A8H
0036' CA ** **            JP     Z,ITMA8
0039'
0039' 3A ** **            LD     A,(ITEM)
003C' E6 7F                      AND     7FH
003E' 26 00                      LD      H,0
0040' 6F                         LD      L,A
0041' 29                         ADD     HL,HL
0042' 11 ** **            LD     DE,CMDTBL
0045' 19                         ADD     HL,DE
0046' 7E                         LD      A,(HL)
0047' 23                         INC     HL
0048' 66                         LD      H,(HL)
0049' 6F                         LD      L,A
004A' E9                         JP      (HL)
004B'
004B' ** ** ** ** **      CMDTBL:        DEFW   ITM80,ITM81,ITM82,ITM83,ITM8
4   ;80
004F' ** ** ** ** **
0053'
0055' ** ** ** ** **             DEFW    ITM85,ITM86,ITM87,ERR0 ,ERR0
0059' ** ** ** ** **
005D'
005F' ** ** ** ** **             DEFW    ERR0 ,ERR0 ,ERR0 ,ERR0 ,ERR0
0063' ** ** ** ** **
0067'
0069' ** **                      DEFW    ERR0
006B' ** ** ** ** **             DEFW    Ierr, Ierr, Ierr, Ierr, Ierr;90
006F' ** ** ** ** **
0073'
0075' ** ** ** ** **             DEFW    Ierr, Ierr, Ierr, Ierr, Ierr
0079' ** ** ** ** **
007D'
007F' ** ** ** ** **             DEFW    Ierr, Ierr, Ierr, Ierr, Ierr
0083' ** ** ** ** **
0087'
0089' ** **                      DEFW    Ierr
008B' ** ** ** ** **             DEFW    Ierr, Ierr, Ierr, Ierr, Ierr;A0
008F' ** ** ** ** **
0093'
0095' ** ** ** ** **             DEFW    Ierr, Ierr, Ierr, Ierr, Ierr
0099' ** ** ** ** **
009D'
009F' ** ** ** ** **             DEFW    Ierr, Ierr, Ierr, Ierr, Ierr
00A3' ** ** ** ** **
00A7'
00A9' ** **                      DEFW    Ierr
00AB' ** ** ** ** **             DEFW    ITMB0,ERR0 ,ERR0 ,ERR0 ,ERR0 ;B0
00AF' ** ** ** ** **
00B3'
00B5' ** ** ** ** **             DEFW    ERR0 ,ERR0 ,ERR0 ,ERR0 ,ERR0
00B9' ** ** ** ** **
00BD'
00BF' ** ** ** ** **             DEFW    ERR0 ,ERR0 ,ERR0 ,ERR0 ,ERR0
00C3' ** ** ** ** **
00C7'
00C9' ** **                      DEFW    ERR0
00CB' ** ** ** ** **             DEFW    ITMC0,ITMC1,ITMC2,ITMC3,ITMC4;C0
00CF' ** ** ** ** **
00D3'
00D5' ** ** ** ** **             DEFW    ITMC5,ITMC6,ITMC7,ITMC8,ITMC9
00D9' ** ** ** ** **
00DD'
00DF' ** ** ** ** **             DEFW    ITMCA,ITMCB,ITMCC,ITMCD,ITMCE
00E3' ** ** ** ** **
00E7'
00E9' ** **                      DEFW    ITMCF
00EB' ** ** ** ** **             DEFW    ERR0 ,ERR0 ,ERR0 ,ERR0 ,ERR0 ;D0
00EF' ** ** ** ** **
00F3'
00F5' ** ** ** ** **             DEFW    ERR0 ,ERR0 ,ERR0 ,ERR0 ,ERR0
00F9' ** ** ** ** **
00FD'
00FF' ** ** ** ** **             DEFW    ERR0 ,ERR0 ,ERR0 ,ERR0 ,ERR0
0103' ** ** ** ** **
0107'
0109' ** **                      DEFW    ERR0
010B' ** ** ** ** **             DEFW    Ierr ,ITME1,ITME2,ITME3,ITME4;E0
010F' ** ** ** ** **
0113'
0115' ** ** ** ** **             DEFW    ITME5,ITME6,ITME7,ITME8,ITME9
0119' ** ** ** ** **
011D'
011F' ** ** ** ** **             DEFW    ERR0 ,ERR0 ,ERR0 ,ERR0 ,ERR0
0123' ** ** ** ** **
0127'
0129' ** **                      DEFW    ERR0
012B' ** ** ** ** **             DEFW    ERR0 ,ERR0 ,ERR0 ,ERR0 ,ERR0 ;F0
012F' ** ** ** ** **
0133'
0135' ** ** ** ** **             DEFW    ERR0 ,ERR0 ,ERR0 ,ERR0 ,ERR0
0139' ** ** ** ** **
013D'
013F' ** ** ** ** **             DEFW    ERR0 ,ITMFB,ITMFC,ITMFD,ITMFE
0143' ** ** ** ** **
0147'
0149' ** **                      DEFW    Ierr
014B'
014B' 11 ** **  Ierr:   LD   DE,MSGI
014E' CD E5 1F          CALL _MSX
0151' 3A ** **          LD   A,(ITEM)
0154' CD C1 1F          CALL _PRTHX
0157' CD EE 1F          CALL _LTNL
015A' C3 FA 1F          JP   _HOT
015D'
015D' 49 6E 74 65 72    MSGI:DB      'Internal Error ITEM-No:',0
0162' 6E 61 6C 20 45
0167' 72 72 6F 72 20
016C' 49 54 45 4D 2D
0171' 4E 6F 3A 00
0175'           •
0175'                   ;
0175'                   ; Define Ext-label
0175'                   ;
0175' CD ** **  ITM00:  CALL GETHL
0178' ED 5B ** **            LD      DE,(LBLOFF)
017C' 19                     ADD     HL,DE
017D'
017D' E5                     PUSH    HL
017E' ED 5B ** **            LD      DE,(NXTOFF)
0182' B7                     OR      A
0183' ED 52                  SBC     HL,DE
0185' E1                     POP     HL
0186' 38 03                  JR      C,ITM001
0188' 22 ** **          LD   (NXTOFF),HL
018B'
018B' E5                ITM001:      PUSH   HL
018C' 3A ** **          LD   A,(ITEM)
018F' E6 1F                  AND     1FH
0191' 47                     LD      B,A
0192' CD ** **          CALL GETLBL
0195' E1                     POP     HL
0196' 3A ** **          LD   A,(pass)
0199' A7                     AND     A
019A' 3E 01                  LD      A,1     ;Certained Label
019C' CC ** **          CALL Z,DEFLBL
019F' C9                     RET
01A0'                   ;
01A0'                   ; Refarende Ext-label
01A0'                   ;
01A0' 3A ** **  ITM20:  LD   A,(ITEM)
01A3' E6 1F                  AND     1FH
01A5' 47                     LD      B,A
01A6' CD ** **          CALL GETLBL
01A9' CD ** **          CALL SEALBL
01AC' 30 18                  JR      NC,ITM201
01AE'
01AE' F5                     PUSH    AF
01AF' CD ** **          CALL GETADRS
```


▶実況，対戦ポピュラス　祝一平 VS 西川善司なんてネタは，たまにOh!Xを見る人にはあまりおもしろくないんだろうなあ。でも，早く読んでみたい。　早川　信平(18)東京都

```
01B2' F1                    POP     AF
01B3' 3E 00                 LD      A,0         ;Uncertained label
01B5' CC ** **          CALL Z,DEFLBL
01B8'
01B8' 3E 01                 LD      A,1
01BA' 32 ** **          LD   (UNDEF),A
01BD' 3A ** **          LD   A,(pass)
01C0' A7                    AND     A
01C1' C4 ** **          CALL NZ,ERR6      ;Undefined Err
01C4' 18 1E                 JR      ITM203
01C6'
01C6' E5                ITM201:     PUSH    HL
01C7' 11 00 60          LD   DE,LBLFLG
01CA' 19                    ADD     HL,DE
01CB' 7E                    LD      A,(HL)
01CC' A7                    AND     A
01CD' E1                    POP     HL
01CE' 20 0C                 JR      NZ,ITM202
01D0'
01D0' 3E 01                 LD      A,1
01D2' 32 ** **          LD   (UNDEF),A
01D5' 3A ** **          LD   A,(pass)
01D8' A7                    AND     A
01D9' C2 ** **          JP   NZ,ERR1      ;Undefined Err
01DC'
01DC' 29                ITM202:     ADD HL,HL
01DD' 11 00 70          LD   DE,LBLNUM
01E0' 19                    ADD     HL,DE
01E1' 5E                    LD      E,(HL)
01E2' 23                    INC     HL
01E3' 56                    LD      D,(HL)
01E4' CD ** **  ITM203: CALL PUSHDE
01E7' C9                    RET
01E8'                   ;
01E8'                   ; Put Object-code
01E8'                   ;
01E8' 3A ** **  ITM60:  LD   A,(ITEM)
01EB' E6 1F                 AND     1FH
01ED' 3C                    INC     A
01EE' 47                    LD      B,A
01EF' C5                ITM601:     PUSH    BC
01F0' CD ** **          CALL INPUT_
01F3' CD ** **          CALL PUTOBJ
01F6' C1                    POP     BC
01F7' 10 F6                 DJNZ    ITM601
01F9' C9                    RET
01FA'                   ;
01FA'                   ; Reference LOC-label
01FA'                   ;
01FA' CD ** **  ITM80:  CALL GETHL
01FD' ED 5B ** **           LD      DE,(LBLOFF)
0201' 19                    ADD     HL,DE
0202'
0202' E5                    PUSH    HL
0203' ED 5B ** **           LD      DE,(NXTOFF)
0207' B7                    OR      A
0208' ED 52                 SBC     HL,DE
020A' E1                    POP     HL
020B' 38 03                 JR      C,ITM801
020D' 22 ** **          LD   (NXTOFF),HL
0210'
0210' E5                ITM801:     PUSH    HL
0211' 11 00 60          LD   DE,LBLFLG
0214' 19                    ADD     HL,DE
0215' 7E                    LD      A,(HL)
0216' E1                    POP     HL
0217' A7                    AND     A
0218' 20 10                 JR      NZ,ITM802    ;This is certained labe

021A'
021A' 3E 01                 LD      A,1
021C' 32 ** **          LD   (UNDEF),A
021F' 3A ** **          LD   A,(pass)
0222' A7                    AND     A
0223' 28 05                 JR      Z,ITM802
0225' CD ** **          CALL ERR1         ;Undefined Label
0228' 18 09                 JR      ITM803
022A'
022A' 11 00 70  ITM802: LD   DE,LBLNUM
022D' 29                    ADD     HL,HL
022E' 19                    ADD     HL,DE
022F' 7E                    LD      A,(HL)
0230' 23                    INC     HL
0231' 66                    LD      H,(HL)
0232' 6F                    LD      L,A
0233' CD ** **  ITM803: CALL PUSHHL
0236' C9                    RET
0237'                   ;
0237'                   ; Define LOC-label
0237'                   ;
0237' CD ** **  ITM81:  CALL GETHL
023A' ED 5B ** **           LD      DE,(LBLOFF)
023E' 19                    ADD     HL,DE
023F'
023F' E5                    PUSH    HL
0240' ED 5B ** **           LD      DE,(NXTOFF)
0244' B7                    OR      A
0245' ED 52                 SBC     HL,DE
0247' E1                    POP     HL
0248' 38 03                 JR      C,ITM811
024A' 22 ** **          LD   (NXTOFF),HL
024D'
024D' E5                ITM811:     PUSH    HL
024E' 11 00 60          LD   DE,LBLFLG
0251' 19                    ADD     HL,DE
0252' 36 01                 LD      (HL),1
0254' E1                    POP     HL
0255'
0255' 29                    ADD     HL,HL
0256' 11 00 70          LD   DE,LBLNUM
0259' 19                    ADD     HL,DE
025A' CD ** **          CALL CPOPDE
025D' 73                    LD      (HL),E
025E' 23                    INC     HL
025F' 72                    LD      (HL),D
0260' C9                    RET
0261'
0261' CD ** **  ITM82:  CALL CPOPHL
0264' E5                    PUSH    HL
0265' 7D                    LD      A,L
0266' CD ** **          CALL PUTOBJ
0269' E1                    POP     HL
026A' 7C                    LD      A,H
026B' CD ** **          CALL PUTOBJ
026E' C9                    RET
026F'
026F' CD ** **  ITM83:  CALL CPOPHL
0272' E5                    PUSH    HL
0273' 7C                    LD      A,H
0274' CD ** **          CALL PUTOBJ
0277' E1                    POP     HL
0278' 7D                    LD      A,L
0279' CD ** **          CALL PUTOBJ
027C' C9                    RET
027D'
027D' CD ** **  ITM84:  CALL CPOPHL
0280' 7D                    LD      A,L
0281' CD ** **          CALL PUTOBJ
0284' C9                    RET
0285'
0285'
0285'
0285' CD ** **  ITM85:  CALL GETADRS
0288' 44                    LD      B,H
0289' 4D                    LD      C,L      ;LD BC,(LOGADR)
028A' CD ** **          CALL CPOPHL      ;LD HL,(PTRxx)
028D'
028D' 3A ** **          LD   A,(pass)
0290' A7                    AND     A
0291' CA ** **          JP   Z,PUTOBJ    ;when pass-1
0294'
0294' 37                    SCF
0295' ED 42                 SBC     HL,BC
0297' 38 0B                 JR      C,JR3
0299' 24                    INC     H
029A' 25                    DEC     H
029B' 20 0F                 JR      NZ,JR4        ;Too Far Err
029D' 7D                    LD      A,L
029E' B7                    OR      A
029F' F2 ** **          JP   P,PUTOBJ
02A2' 18 08                 JR      JR4      ;Too Far
02A4' 24                JR3: INC    H
02A5' 20 05                 JR      NZ,JR4
02A7' 7D                    LD      A,L
02A8' B7                    OR      A
02A9' FA ** **          JP   M,PUTOBJ
02AC' CD ** **  JR4:    CALL PUTOBJ
02AF' C3 ** **          JP   ERR4         ;Too Far Err
02B2'
02B2' CD ** **  ITM86:  CALL GETADRS
02B5' E5                    PUSH    HL
02B6' CD ** **          CALL GETHL
02B9' D1                    POP     DE
02BA' 19                    ADD     HL,DE
02BB' CD ** **          CALL PUTHL
02BE' C9                    RET
02BF'
02BF' CD ** **  ITM87:  CALL GETADRS
02C2' CD ** **          CALL PUSHHL
02C5' C9                    RET
02C6'
02C6' 3A ** **  ITM90:  LD   A,(ITEM)
02C9' E6 07                 AND     7
02CB' 3C                    INC     A
02CC' 47                    LD      B,A
02CD' C5                ITM901:     PUSH    BC
02CE' CD ** **          CALL INPUT_
02D1' CD ** **          CALL PUTOBJ
02D4' C1                    POP     BC
02D5' 10 F6                 DJNZ    ITM901
02D7' C9                    RET
02D8'
02D8' 3A ** **  ITM98:  LD   A,(ITEM)
02DB' E6 07                 AND     7
02DD' 3C                    INC     A
02DE' 47                    LD      B,A
02DF' C5                ITM981:     PUSH    BC
02E0' CD ** **          CALL GETHL
02E3' CD ** **          CALL PUTHL
02E6' C1                    POP     BC
02E7' 10 F6                 DJNZ    ITM981
02E9' C9                    RET
02EA'
02EA' 3A ** **  ITMA0:  LD   A,(ITEM)
02ED' E6 07                 AND     7
02EF' 3C                    INC     A
02F0' 47                    LD      B,A
02F1' C5                ITMA01:     PUSH    BC
02F2' CD ** **          CALL GETHL
02F5' E5                    PUSH    HL
02F6' 7C                    LD      A,H
02F7' CD ** **          CALL PUTOBJ
02FA' E1                    POP     HL
02FB' 7D                    LD      A,L
02FC' CD ** **          CALL PUTOBJ
02FF' C1                    POP     BC
0300' 10 EF                 DJNZ    ITMA01
0302' C9                    RET
0303'
0303' C3 ** **  ITMA8:  JP   ITM98
0306'
0306' CD ** **  ITMB0:  CALL CPOPHL
0309' 7C                ITMB01:     LD  A,H
030A' B5                    OR      L
030B' C8                    RET     Z
030C' E5                    PUSH    HL
030D' AF                    XOR     A
030E' CD ** **          CALL PUTOBJ
0311' E1                    POP     HL
0312' 2B                    DEC     HL
0313' 18 F4                 JR      ITMB01
0315'
0315' CD ** **  ITMC0:  CALL POPDE
0318' CD ** **          CALL POPHL
031B' 19                    ADD     HL,DE
031C' CD ** **          CALL PUSHHL
031F' C9                    RET
0320'
0320' CD ** **  ITMC1:  CALL POPDE
0323' CD ** **          CALL POPHL
0326' B7                    OR      A
0327' ED 52                 SBC     HL,DE
0329' CD ** **          CALL PUSHHL
032C' C9                    RET
032D'
032D' CD ** **  ITMC2:  CALL POPDE
0330' CD ** **          CALL POPHL
0333' CD ** **          CALL MUL
0336' CD ** **          CALL PUSHHL
0339' C9                    RET
033A'
033A' CD ** **  ITMC3:  CALL POPDE
033D' CD ** **          CALL POPHL
0340' CD ** **          CALL DIV
0343' CD ** **          CALL PUSHHL
0346' C9                    RET
0347'
0347' CD ** **  ITMC4:  CALL POPHL
034A' 6C                    LD      L,H
034B' 26 00                 LD      H,0
034D' CD ** **          CALL PUSHHL
0350' C9                    RET
0351'
0351' CD ** **  ITMC5:  CALL POPHL
0354' 26 00                 LD      H,0
0356' CD ** **          CALL PUSHHL
0359' C9                    RET
035A'
035A' CD ** **  ITMC6:  CALL POPHL
035D' 7C                    LD      A,H
035E' 2F                    CPL
035F' 67                    LD      H,A
0360' 7D                    LD      A,L
0361' 2F                    CPL
0362' 6F                    LD      L,A
0363' 23                    INC     HL
0364' CD ** **          CALL PUSHHL
0367' C9                    RET
0368'
0368' CD ** **  ITMC7:  CALL POPDE
036B' CD ** **          CALL POPHL
036E' CD ** **          CALL DIV
0371' EB                    EX      DE,HL
0372' CD ** **          CALL PUSHHL
0375' C9                    RET
0376'
0376' CD ** **  ITMC8:  CALL POPHL
0379' CD ** **          CALL POPDE
037C'
037C' 7A                    LD      A,D
037D' B3                    OR      E
037E' 28 13                 JR      Z,ITMC82     ;DE = 0→ PUSH(1)
0380'
0380' 42                    LD      B,D
0381' 4B                    LD      C,E     ;BC = DE
0382' 54                    LD      D,H
0383' 5D                    LD      E,L     ;DE = HL
0384'
0384' 0B                ITMC81:     DEC BC
0385' 78                    LD      A,B
0386' B1                    OR      C
0387' CA ** **          JP   Z,PUSHHL
038A'
038A' C5                    PUSH    BC
038B' D5                    PUSH    DE
038C' CD ** **          CALL MUL        ;HL = HL* DE
038F' D1                    POP     DE
0390' C1                    POP     BC
0391' 18 F1                 JR      ITMC81
0393'
0393' 21 01 00  ITMC82: LD   HL,1       ; HL ^ 0 = 1
0396' CD ** **          CALL PUSHHL
0399' C9                    RET
039A'
039A' CD ** **  ITMC9:  CALL POPDE
039D' CD ** **          CALL POPHL
03A0' CB 3C             ITMC91:     SRL H
03A2' CB 1D                 RR      L
03A4' 1D                    DEC     E
03A5' 20 F9                 JR      NZ,ITMC91
03A7' CD ** **          CALL PUSHHL
03AA' C9                    RET
03AB'
03AB' CD ** **  ITMCA:  CALL POPDE
03AE' CD ** **          CALL POPHL
03B1' 29                ITMCA1:     ADD HL,HL
03B2' 1D                    DEC     E
03B3' 20 FC                 JR      NZ,ITMCA1
03B5' CD ** **          CALL PUSHHL
03B8' C9                    RET
03B9'
03B9' CD ** **  ITMCB:  CALL POPHL
03BC' 7C                    LD      A,H
03BD' 2F                    CPL
03BE' 67                    LD      H,A
03BF' 7D                    LD      A,L
03C0' 2F                    CPL
03C1' 6F                    LD      L,A
03C2' CD ** **          CALL PUSHHL
03C5' C9                    RET
03C6'
03C6' CD ** **  ITMCC:  CALL POPDE
03C9' CD ** **          CALL POPHL
03CC' 7C                    LD      A,H
03CD' B2                    OR      D
03CE' 67                    LD      H,A
03CF' 7D                    LD      A,L
03D0' B3                    OR      E
03D1' 6F                    LD      L,A
03D2' CD ** **          CALL PUSHHL
03D5' C9                    RET
03D6'
03D6' CD ** **  ITMCD:  CALL POPDE
03D9' CD ** **          CALL POPHL
03DC' 7C                    LD      A,H
03DD' B2                    OR      D
03DE' 67                    LD      H,A
03DF' 7D                    LD      A,L
03E0' B3                    OR      E
03E1' 6F                    LD      L,A
03E2' CD ** **          CALL PUSHHL
03E5' C9                    RET
03E6'
03E6' CD ** **  ITMCE:  CALL POPDE
03E9' CD ** **          CALL POPHL
03EC' 7C                    LD      A,H
03ED' AA                    XOR     D
03EE' 67                    LD      H,A
03EF' 7D                    LD      A,L
03F0' AB                    XOR     E
03F1' 6F                    LD      L,A
03F2' CD ** **          CALL PUSHHL
03F5' C9                    RET
03F6'
03F6' CD ** **  ITMCF:  CALL POPHL
03F9' 7C                    LD      A,H
03FA' 65                    LD      H,L
03FB' 6F                    LD      L,A
03FC' CD ** **          CALL PUSHHL
03FF' C9                    RET
0400'
0400' CD ** **  ITME1:  CALL INPUT_
0403' 06 02                 LD      B,2     ;CSEG
0405' A7                    AND     A
0406' 28 07                 JR      Z,ITME11
0408' 06 03                 LD      B,3     ;CSEG
040A' 3D                    DEC     A
040B' 28 02                 JR      Z,ITME11
040D' 06 04                 LD      B,4     ;WSEG
040F' 78                ITME11:     LD  A,B
0410' 32 ** **          LD   (SEGMD),A
0413' C9                    RET
0414'
0414' CD ** **  ITME2:  CALL CPOPHL
0417' 22 ** **          LD   (PTRFC),HL
041A' 3E 01                 LD      A,1
041C' 32 ** **          LD   (LOCFLG),A
041F' C9                    RET
0420'
0420' AF                ITME3:      XOR A
0421' 32 ** **          LD   (LOCFLG),A
0424' C9                    RET
0425'
0425' CD ** **  ITME4:  CALL CPOPHL
0428' 7D                    LD      A,L
0429' CD ** **          CALL PUTOBJ
042C' C9                    RET
042D'
042D' CD ** **  ITME5:  CALL CPOPHL
0430' CD ** **          CALL PUTHL
0433' C9                    RET
0434'
0434' CD ** **  ITME6:  CALL CPOPHL
0437' E5                    PUSH    HL
0438' 7C                    LD      A,H
0439' CD ** **          CALL PUTOBJ
043C' E1                    POP     HL
043D' 7D                    LD      A,L
043E' CD ** **          CALL PUTOBJ
0441' C9                    RET
0442'
0442' CD ** **  ITME7:  CALL GETHL
0445' CD ** **          CALL PUSHHL
0448' C9                    RET
0449'
0449' CD ** **  ITME8:  CALL GETADRS
044C' EB                    EX      DE,HL       ;DE = Current Address
044D' CD ** **          CALL CPOPHL     ;HL = Objective Address
0450' B7                    OR      A
0451' ED 52                 SBC     HL,DE
0453' C8                    RET     Z
0454' DA ** **          JP   C,ERR9     ;Illegal ORG Err
0457' E5                ITME81:     PUSH    HL
0458' AF                    XOR     A
0459' CD ** **          CALL PUTOBJ     ;Put Dummy Data to adjust PC
045C' E1                    POP     HL
045D' 2B                    DEC     HL
045E' 7C                    LD      A,H
045F' B5                    OR      L
0460' 20 F5                 JR      NZ,ITME81
0462' C9                    RET
0463'
0463' CD ** **  ITME9:  CALL POPHL
0466' 22 ** **          LD   (EXADR),HL
0469' 3E 01                 LD      A,1
046B' 32 ** **          LD   (FLGEX),A
046E' C9                    RET
046F'
046F' CD ** **  ITMFB:  CALL INPUT_
0472' A7                    AND     A
0473' C8                    RET     Z
0474' CD F4 1F          CALL _PRINT
0477' 18 F6                 JR      ITMFB
0479'
0479' CD ** **  ITMFC:  CALL POPHL
047C' CD ** **          CALL PRDEC
047F' C9                    RET
0480'
0480' CD ** **  ITMFD:  CALL POPHL
0483' CD BE 1F          CALL _PRTHL
0486' C9                    RET
0487'
0487' CD ** **  ITMFE:  CALL POPHL
048A' 7D                    LD      A,L
048B' CD C1 1F          CALL _PRTHX
048E' C9                    RET
048F'
048F'
048F'                   ;
048F'                   ; Define a label
048F'                   ;
048F' 22 ** **  DEFLBL: LD   (SV_VAR),HL
0492' 32 ** **          LD   (SV_FLG),A
0495'
0495' CD ** **          CALL SEALBL
0498' D2 ** **          JP   NC,ERR5     ;Multi Defined Err
049B' 20 42                 JR      NZ,DEF3
049D'
049D' CD ** **          CALL HASH
04A0' 26 00                 LD      H,0
04A2' 6F                    LD      L,A
04A3' 29                    ADD     HL,HL        ;HL = HASH * 2
04A4'
04A4' 44                    LD      B,H
04A5' 4D                    LD      C,L     ;BC = LABEL BUFFER POINTER
04A6'
04A6' 60                DEF1:LD     H,B
04A7' 69                    LD      L,C
04A8' CD ** **          CALL PEEK_BC
```


▶ぜんまいちゃんと聞いてなつかしさを感じたのは友達のなかではぼくだけでしょう。あのころはよかった。今もいいけど……。
鈴木　武虎(16)愛知県

```
04AB' 78                        LD      A,B
04AC' B1                        OR      C
04AD' 20 F7                     JR      NZ,DEF1
04AF'
04AF'                   ; Chain Last struct with New struct
04AF' 2B                        DEC     HL
04B0' 2B                        DEC     HL
04B1' ED 4B ** **               LD      BC,(LBLPTR)
04B5' CD ** **          CALL POKE_BC
04B8'
04B8'                   ;** Make Struct of Labels **
04B8' 2A ** **          LD   HL,(LBLPTR)
04BB' 01 00 00          LD   BC,0
04BE' CD ** **          CALL POKE_BC
04C1' 3A ** **          LD   A,(SV_FLG)
04C4' CD ** **          CALL POKE_I
04C7' ED 4B ** **               LD      BC,(SV_VAR)
04CB' CD ** **          CALL POKE_BC
04CE'
04CE' 11 ** **          LD   DE,LBLBUF
04D1' 1A                DEF2:LD         A,(DE)
04D2' 13                        INC     DE
04D3' F5                        PUSH    AF
04D4' CD ** **          CALL POKE_I
04D7' F1                        POP     AF
04D8' A7                        AND     A
04D9' 20 F6                     JR      NZ,DEF2
04DB'
04DB' 22 ** **          LD   (LBLPTR),HL
04DE'                   ;       OR      A       ;CY = 0
04DE' C9                        RET
04DF'
04DF' 2A ** **  DEF3:   LD   HL,(SV_STR)
04E2' 23                        INC     HL
04E3' 23                        INC     HL
04E4' 3A ** **          LD   A,(SV_FLG)
04E7' CD ** **          CALL POKE_I
04EA' ED 4B ** **               LD      BC,(SV_VAR)
04EE' CD ** **          CALL POKE_BC
04F1' B7                        OR      A       ;CY = 0
04F2' C9                        RET
04F3'
04F3'
04F3'                   SV_VAR:         DS  2
04F5'                   SV_FLG:         DS  1
04F6'                   ;
04F6'                   ;       Search Label
04F6'                   ;
04F6' CD ** **  SEALBL::CALL HASH
04F9' 26 00                     LD      H,0
04FB' 6F                        LD      L,A
04FC' 29                        ADD     HL,HL
04FD' 22 ** **          LD   (SV_STR),HL
0500'
0500' 2A ** **  SEA1:   LD   HL,(SV_STR)
0503' CD ** **          CALL PEEK_BC
0506' 78                        LD      A,B
0507' B1                        OR      C
0508' 37                        SCF
0509' C8                        RET     Z       ;Not Defined (CY = 1)
050A'
050A' ED 43 ** **               LD      (SV_STR),BC
050E' 21 05 00          LD   HL,5
0511' 09                        ADD     HL,BC           ;HL = &(Label->str)
0512'
0512'                   ; Check The Labels are the same
0512'
0512' 11 ** **          LD   DE,LBLBUF
0515' 1A                SEA2:LD         A,(DE)
0516' A7                        AND     A
0517' 28 0A                     JR      Z,SEA3          ;The Same !!
0519' 13                        INC     DE
051A' 47                        LD      B,A
051B' CD ** **          CALL PEEK_I
051E' B8                        CP      B
051F' 28 F4                     JR      Z,SEA2
0521'
0521' 18 DD                     JR      SEA1            ;Not The Same
0523'
0523' CD 94 1F  SEA3:   CALL _PEEK
0526' A7                        AND     A       ;END CODE ?
0527' 20 D7                     JR      NZ,SEA1         ;Not The Same
0529'
0529' 2A ** **          LD   HL,(SV_STR)
052C' 23                        INC     HL
052D' 23                        INC     HL
052E' CD ** **          CALL PEEK_I     ;A = label flag
0531' CD ** **          CALL PEEK_BC    ;BC= the value
0534' 60                        LD      H,B
0535' 69                        LD      L,C
0536' B7                        OR      A
0537' C0                        RET     NZ      ;CY = 0
0538' 3E 01                     LD      A,1
053A' B7                        OR      A
053B' 37                        SCF
053C' C9                        RET             ;CY = 1 & Z = 0
053D'
053D'                   SV_STR:         DS  2
053F'
053F'
053F'
053F' 21 ** **  GETLBL: LD   HL,LBLBUF
0542' C5                GETLB1:         PUSH    BC
0543' E5                        PUSH    HL
0544' CD ** **          CALL INPUT_
0547' E1                        POP     HL
0548' C1                        POP     BC
0549' 77                        LD      (HL),A
054A' 23                        INC     HL
054B' 10 F5                     DJNZ    GETLB1
054D' 36 00                     LD      (HL),0
054F' C9                        RET
0550'
0550'
0550' E5                HASH:PUSH       HL
0551' C5                        PUSH    BC
0552' 21 ** **          LD   HL,LBLBUF
0555' 06 00                     LD      B,0
0557' 7E                HASH1:          LD  A,(HL)
0558' 23                        INC     HL
0559' A7                        AND     A
055A' 28 04                     JR      Z,HASH2
055C' 80                        ADD     A,B
055D' 47                        LD      B,A
055E' 18 F7                     JR      HASH1
0560' 78                HASH2:          LD  A,B
0561' C1                        POP     BC
0562' E1                        POP     HL
0563' C9                        RET
0564'                   ;
0564'                   ; Init hash table
0564'                   ;
0564' 21 00 00  INIHSH::LD   HL,0
0567' 01 01 02          LD   BC,200H+1
056A' AF                        XOR     A
056B' CD 9A 1F  INIHS0: CALL _POKE
056E' ED A1                     CPI
0570' EA ** **          JP   PE,INIHS0
0573' C9                        RET
0574'
0574'                   ;
0574'                   ; Put EXT-Undefined LABEL to CRT
0574'                   ;
0574' AF                PRTMAP::XOR  A
0575' 32 ** **          LD   (PRTLOC),A
0578' 21 00 00          LD   HL,0
057B' 22 ** **          LD   (PRTCNT),HL
057E' 21 00 02          LD   HL,200H
0581' ED 5B ** **       PTMP1:          LD  DE,(LBLPTR)
0585' E5                        PUSH    HL
0586' B7                        OR      A
0587' ED 52                     SBC     HL,DE
0589' E1                        POP     HL
058A' 30 14                     JR      NC,PTMP6
058C'
058C' 23                        INC     HL
058D' 23                        INC     HL
058E' CD ** **          CALL PEEK_I
0591' A7                        AND     A
0592' 20 05                     JR      NZ,PTMP5
0594' CD ** **          CALL PRTname
0597' 18 E8                     JR      PTMP1
0599'
0599' 23                PTMP5:          INC HL
059A' 23                        INC     HL
059B' CD ** **          CALL SKPname
059E' 18 E1                     JR      PTMP1
05A0'
05A0' CD EB 1F  PTMP6:  CALL _NL
05A3' 2A ** **          LD   HL,(PRTCNT)
05A6' 7C                        LD      A,H
05A7' B5                        OR      L
05A8' C8                        RET     Z
05A9' CD ** **          CALL PRDEC
05AC' 21 ** **          LD   HL,MSGUN
05AF' CD ** **          CALL _puts
05B2' C9                        RET
05B3'
05B3'
05B3' 20 55 6E 64 65    MSGUN:          DB  ' Undefined Label(s)',CR,0
05B8' 66 69 6E 65 64
05BD' 20 4C 61 62 65
05C2' 6C 28 73 29 0D
05C7' 00
05C8'
05C8' 3E 2D             PRTname:LD   A,'-'
05CA' CD F4 1F          CALL _PRINT
05CD' CD ** **          CALL PEEK_BC
05D0' C5                        PUSH    BC
05D1' 06 0A                     LD      B,10            ;Max length of printed,
label
05D3' CD ** **  PTMP2:  CALL PEEK_I
05D6' A7                        AND     A
05D7' 28 08                     JR      Z,PTMP3
05D9' CD F4 1F          CALL _PRINT
05DC' 10 F5                     DJNZ    PTMP2
05DE' CD ** **          CALL SKPname
05E1'
05E1' 04                PTMP3:          INC B
05E2' CD F1 1F  PTMP4:  CALL _PRNTS
05E5' 10 FB                     DJNZ    PTMP4
05E7'
05E7' E3                        EX      (SP),HL
05E8' CD BE 1F          CALL _PRTHL
05EB' CD F1 1F          CALL _PRNTS
05EE' 2A ** **          LD   HL,(PRTCNT)
05F1' 23                        INC     HL
05F2' 22 ** **          LD   (PRTCNT),HL
05F5' 3A ** **          LD   A,(PRTLOC)
05F8' 3C                        INC     A
05F9' E6 03                     AND     3
05FB' 32 ** **          LD   (PRTLOC),A
05FE' CC EB 1F          CALL Z,_NL
0601' E1                        POP     HL
0602' C9                        RET
0603'
0603' CD ** **  SKPname:CALL PEEK_I
0606' A7                        AND     A
0607' C8                        RET     Z
0608' 18 F9                     JR      SKPname
060A'
060A'                   ;
060A'                   ; Put Cross Reference Data
060A'                   ;     into the MAP-file
060A'                   ;
060A' AF                PUTMAP::XOR  A
060B' 32 ** **          LD   (PRTLOC),A
060E' 21 00 02          LD   HL,200H
0611'
0611' E5                PUTM1:          PUSH    HL
0612' ED 5B ** **               LD      DE,(LBLPTR)
0616' B7                        OR      A
0617' ED 52                     SBC     HL,DE
0619' E1                        POP     HL
061A' D2 ** **          JP   NC,PUTM7
061D'
061D' 23                        INC     HL
061E' 23                        INC     HL
061F'
061F' 3E 2D                     LD      A,'-'
0621' CD ** **          CALL PRT_
0624'
0624' CD ** **          CALL PEEK_I
0627' A7                        AND     A
0628' 28 1A                     JR      Z,PUTM4
062A'
062A' CD ** **          CALL PEEK_BC
062D' C5                        PUSH    BC
062E' CD ** **          CALL PUTname
0631' E3                PUTM3:          EX  (SP),HL
0632' 29                        ADD     HL,HL
0633' 11 00 70          LD   DE,LBLNUM
0636' 19                        ADD     HL,DE
0637' 7E                        LD      A,(HL)
0638' 23                        INC     HL
0639' 66                        LD      H,(HL)
063A' 6F                        LD      L,A
063B' CD ** **          CALL PUTmHL
063E' CD ** **          CALL PUTmCR
0641' E1                        POP     HL
0642' 18 CD                     JR      PUTM1
0644'
0644' 23                PUTM4:          INC HL
0645' 23                        INC     HL
0646' CD ** **  PUTM5:  CALL PUTname
0649' 11 ** **  PUTM6:  LD   DE,PUTmm
064C' CD ** **          CALL PUTmsg
064F' CD ** **          CALL PUTmCR
0652' C3 ** **          JP   PUTM1
0655'
0655' 3E 0D             PUTM7:          LD  A,CR
0657' CD ** **          CALL PRT_
065A' AF                        XOR     A
065B' CD ** **          CALL PRT_
065E' C9                        RET
065F'
065F' 2A 2A 2A 2A 00    PUTmm:          DB  '****',0
0664'
0664'
0664'
0664' 06 0A             PUTname:LD   B,10               ;Max length of printed
labels
0666' CD ** **  PUTn1:  CALL PEEK_I
0669' A7                        AND     A
066A' 28 08                     JR      Z,PUTn2
066C' CD ** **          CALL PRT_
066F' 10 F5                     DJNZ    PUTn1
0671' CD ** **          CALL SKPname
0674'
0674' 04                PUTn2:          INC B
0675' 3E 20             PUTn3:          LD  A,
0677' CD ** **          CALL PRT_
067A' 10 F9                     DJNZ    PUTn3
067C' C9                        RET
067D'
067D' 3A ** **  PUTmCR: LD   A,(PRTLOC)
0680' 3C                        INC     A
0681' E6 03                     AND     3
0683' 32 ** **          LD   (PRTLOC),A
0686' 3E 20                     LD      A,'
0688' C2 ** **          JP   NZ,PRT_
068B' 3E 0D                     LD      A,CR
068D' C3 ** **          JP   PRT_
0690'
0690'
0690' 1A                PUTmsg:         LD  A,(DE)
0691' A7                        AND     A
0692' C8                        RET     Z
0693' 13                        INC     DE
0694' CD ** **          CALL PRT_
0697' 18 F7                     JR      PUTmsg
0699'
0699' 7C                PUTmHL:         LD  A,H
069A' CD ** **          CALL PUTmHX
069D' 7D                        LD      A,L
069E' F5                PUTmHX:         PUSH    AF
069F' 0F                        RRCA
06A0' 0F                        RRCA
06A1' 0F                        RRCA
06A2' 0F                        RRCA
06A3' CD ** **          CALL PRTHX1
06A6' F1                        POP     AF
06A7' CD ** **  PRTHX1: CALL ASC
06AA' 18 0A                     JR      PRT_
06AC'
06AC' E6 0F             ASC: AND        00FH
06AE' F6 30                     OR      030H
06B0' FE 3A                     CP      3AH
06B2' D8                        RET     C
06B3' C6 07                     ADD     A,7
06B5' C9                        RET
06B6'
06B6' C5                PRT_:PUSH       BC
06B7' D5                        PUSH    DE
06B8' E5                        PUSH    HL
06B9' CD ** **          CALL PRINT_
06BC' E1                        POP     HL
06BD' D1                        POP     DE
06BE' C1                        POP     BC
06BF' C9                        RET
06C0'
06C0'
06C0'                   ;
06C0'                   ; Put HL data to the object-file
06C0'                   ;
06C0' E5                PUTHL:          PUSH    HL
06C1' 7D                        LD      A,L
06C2' CD ** **          CALL PUTOBJ
06C5' E1                        POP     HL
06C6' 7C                        LD      A,H
06C7' CD ** **          CALL PUTOBJ
06CA' C9                        RET
06CB'
06CB' D5                PUSHHL:         PUSH    DE
06CC' 54                        LD      D,H
06CD' 5D                        LD      E,L
06CE' CD ** **          CALL PUSHDE
06D1' D1                        POP     DE
06D2' C9                        RET
06D3'
06D3' 3A ** **  PUSHDE: LD   A,(STLVL)
06D6' FE 11                     CP      STMAX + 1
06D8' D2 ** **          JP   NC,ERR7    ;Stack Overflow
06DB'
06DB' E5                        PUSH    HL
06DC' D5                        PUSH    DE
06DD' 26 00                     LD      H,0
06DF' 6F                        LD      L,A
06E0' 29                        ADD     HL,HL
06E1' 11 ** **          LD   DE,STBUF
06E4' 19                        ADD     HL,DE           ;HL = STBUF + (STLVL) *
2
06E5' D1                        POP     DE
06E6' 73                        LD      (HL),E
06E7' 23                        INC     HL
06E8' 72                        LD      (HL),D
06E9' E1                        POP     HL
06EA' 3C                        INC     A
06EB' 32 ** **          LD   (STLVL),A
06EE' C9                        RET
06EF'
06EF'                   ;
06EF'                   ; Clear (UNDEF) & POP DE
06EF'                   ;
06EF' AF                CPOPDE:         XOR A
06F0' 32 ** **          LD   (UNDEF),A
06F3' E5                POPDE:          PUSH    HL
06F4' CD ** **          CALL POPHL
06F7' 54                        LD      D,H
06F8' 5D                        LD      E,L
06F9' E1                        POP     HL
06FA' C9                        RET
06FB'
06FB'                   ;
06FB'                   ; Clear (UNDEF) & POP HL
06FB'                   ;
06FB' AF                CPOPHL:         XOR A
06FC' 32 ** **          LD   (UNDEF),A
06FF'                   ;       CALL    POPHL
06FF'                   ;       RET
06FF'
06FF'                   ;
06FF'                   ; POP HL from EV-stack
06FF'                   ;
06FF' 3A ** **  POPHL:  LD   A,(STLVL)
0702' A7                        AND     A
0703' CA ** **          JP   Z,ERR8     ;Stack Empty
0706' 3D                        DEC     A
0707' 32 ** **          LD   (STLVL),A
070A'
070A' D5                        PUSH    DE
070B' 26 00                     LD      H,0
070D' 6F                        LD      L,A
070E' 29                        ADD     HL,HL
070F' 11 ** **          LD   DE,STBUF
0712' 19                        ADD     HL,DE           ;HL = STBUF + (STLVL) *
2
0713' 7E                        LD      A,(HL)
0714' 23                        INC     HL
0715' 66                        LD      H,(HL)
0716' 6F                        LD      L,A
0717' D1                        POP     DE
0718' C9                        RET
0719'                   ;
0719'                   ; PUT into OBJ-file from A reg.
0719'                   ;
0719' CD ** **  PUTOBJ: CALL INCPC
071C' 26 00                     LD      H,0
071E' 6F                        LD      L,A
071F'
071F' 3A ** **          LD   A,(pass)
0722' A7                        AND     A
0723' C8                        RET     Z       ;RET,when pass-1
0724'
0724' 3A ** **          LD   A,(SEGMD)
0727' FE 03                     CP      3
0729' 28 0C                     JR      Z,PUTOBJ1
072B' FE 04                     CP      4
072D' C8                        RET     Z       ;WSEG
072E'                   ;CSEG
072E' 3A ** **          LD   A,(FLGCD)
0731' A7                        AND     A
0732' 7D                        LD      A,L
0733' C4 ** **          CALL NZ,PRINT_
0736' C9                        RET
0737'
0737' 3A ** **  PUTOBJ1:LD   A,(FLGDT)  ;DSEG
073A' A7                        AND     A
073B' C8                        RET     Z
073C'
073C' E5                        PUSH    HL
073D' 2A ** **          LD   HL,(CNTDT)
0740' 23                        INC     HL
0741' 22 ** **          LD   (CNTDT),HL
0744' 7C                        LD      A,H
0745' 11 00 90          LD   DE,BF_DSEG
0748' 19                        ADD     HL,DE
0749' D1                        POP     DE
074A' FE 20                     CP      1FH + 1
074C' 30 02                     JR      NC,PUTOBJ2
074E' 73                        LD      (HL),E
074F' C9                        RET
0750'
0750' 21 ** **  PUTOBJ2:LD   HL,PUTOBJ3
0753' CD ** **          CALL _puts
0756' C3 ** **          JP   exit
0759' 44 53 45 47 20    PUTOBJ3:DB   'DSEG buffer is over !!',CR,0
075E' 62 75 66 66 65
0763' 72 20 69 73 20
0768' 6F 76 65 72 20
076D' 21 21 0D 00
0771'
0771'                   ;
0771'                   ; GET 2 byte from REL-file INTO HL
0771'                   ;
0771' CD ** **  GETHL:  CALL INPUT_
0774' F5                        PUSH    AF
0775' CD ** **          CALL INPUT_
0778' 67                        LD      H,A
0779' F1                        POP     AF
077A' 6F                        LD      L,A
077B' C9                        RET
077C'
077C'                   ;
077C'                   ; GET ADDRESS
077C'                   ;
077C' 2A ** **  GETADRS:LD   HL,(PTRFC)
077F' 3A ** **          LD   A,(LOCFLG)
0782' A7                        AND     A
```

▶M（メガ：10^{6}），G（ギガ：10^{9}）ときて，そのあとはT（テラ：10^{12}），P（ペタ：10^{15}），E（エクサ：10^{18}）だそーです。1Eバイト＝1兆995億1162万7776Mバイトって計算だぁ！　使い切れんな……。

小藪　賢(21)埼玉県

```
0783' C0                    RET     NZ
0784' 2A ** **         LD   HL,(PTRCD)
0787' 3A ** **         LD   A,(SEGMD)
078A' FE 02                 CP      2
078C' C8                    RET     Z
078D' 2A ** **         LD   HL,(PTRDT)
0790' FE 03                 CP      3
0792' C8                    RET     Z
0793' 2A ** **         LD   HL,(PTRWK)
0796' C9                    RET
0797'
0797'                  ;
0797'                  ; Increment respectable PC
0797'                  ;
0797' F5               INCPC:       PUSH    AF
0798' E5                    PUSH    HL
0799' 3A ** **         LD   A,(SEGMD)
079C' FE 03                 CP      3
079E' 28 0D                 JR      Z,INCpc1
07A0' FE 04                 CP      4
07A2' 28 12                 JR      Z,INCpc2
07A4'
07A4' 2A ** **         LD   HL,(PTRCD)  ;CSEG
07A7' 23                    INC     HL
07A8' 22 ** **         LD   (PTRCD),HL
07AB' 18 10                 JR      INCpc3
07AD'
07AD' 2A ** **  INCpc1: LD   HL,(PTRDT)  ;DSEG
07B0' 23                    INC     HL
07B1' 22 ** **         LD   (PTRDT),HL
07B4' 18 07                 JR      INCpc3
07B6'
07B6' 2A ** **  INCpc2: LD   HL,(PTRWK)  ;WSEG
07B9' 23                    INC     HL
07BA' 22 ** **         LD   (PTRWK),HL
07BD'
07BD' 2A ** **  INCpc3: LD   HL,(PTRFC)
07C0' 23                    INC     HL
07C1' 22 ** **         LD   (PTRFC),HL
07C4' E1                    POP     HL
07C5' F1                    POP     AF
07C6' C9                    RET
07C7'                  ;
07C7'                  ; Error Routines
07C7'                  ;
07C7' 11 ** **  ERR0:  LD   DE,MSG0
07CA' CD E5 1F         CALL _MSX
07CD' 3A ** **         LD   A,(ITEM)
07D0' CD C1 1F         CALL _PRTHX
07D3' CD EE 1F         CALL _LTNL
07D6' C9                    RET
07D7'
07D7' E5               ERR1:PUSH    HL
07D8' 11 ** **         LD   DE,MSG1
07DB' CD E5 1F         CALL _MSX
07DE' E1                    POP     HL
07DF' CD BE 1F         CALL _PRTHL
07E2' CD EE 1F         CALL _LTNL
07E5' C9                    RET
07E6'
07E6' E5               ERR2:PUSH    HL
07E7' 11 ** **         LD   DE,MSG2
07EA' CD E5 1F         CALL _MSX
07ED' E1                    POP     HL
07EE' CD BE 1F         CALL _PRTHL
07F1' CD EE 1F         CALL _LTNL
07F4' C9                    RET
07F5'
07F5' 11 ** **  ERR3:  LD   DE,MSG3
07F8' C3 ** **         JP   ERROR
07FB'
07FB' 11 ** **  ERR4:  LD   DE,MSG4
07FE' C3 ** **         JP   ERROR
0801'
0801' 11 ** **  ERR5:  LD   DE,MSG5
0804' CD E5 1F         CALL _MSX
0807' 11 ** **         LD   DE,LBLBUF
080A' 18 1A                 JR      ERROR
080C'
080C' 11 ** **  ERR6:  LD   DE,MSG6
080F' CD E5 1F         CALL _MSX
0812' 11 ** **         LD   DE,LBLBUF
0815' 18 0F                 JR      ERROR
0817'
0817' 11 ** **  ERR7:  LD   DE,MSG7
081A' 18 0A                 JR      ERROR
081C'
081C' 11 ** **  ERR8:  LD   DE,MSG8
081F' 18 05                 JR      ERROR
0821'
0821' 11 ** **  ERR9:  LD   DE,MSG9
0824' 18 00                 JR      ERROR
0826'
0826' CD E5 1F  ERROR: CALL _MSX
0829' CD EE 1F         CALL _LTNL
082C' C9                    RET
082D'
082D' 55 6E 64 65 66   MSG0:DB      'Undefined ITEM',0
0832' 69 6E 65 64 20
0837' 49 54 45 4D 00
083C'
083C' 55 6E 64 65 66   MSG1:DB      'Undefined Label-No',0
0841' 69 6E 65 64 20
0846' 4C 61 62 65 6C
084B' 2D 4E 6F 00
084F' 4D 75 6C 74 69   MSG2:DB      'Multi Defined Label-No',0
0854' 20 44 65 66 69
0859' 6E 65 64 20 4C
085E' 61 62 65 6C 2D
0863' 4E 6F 00
0866' 54 6F 6F 20 4D   MSG3:DB      'Too Many Labels',0
086B' 61 6E 79 20 4C
0870' 61 62 65 6C 73
0875' 00
0876' 54 6F 6F 20 46   MSG4:DB      'Too Far',0
087B' 61 72 00
087E' 4D 75 6C 74 69   MSG5:DB      'Multi Defined Label-',0
0883' 20 44 65 66 69
0888' 6E 65 64 20 4C
088D' 61 62 65 6C 2D
0892' 00
0893' 55 6E 64 65 66   MSG6:DB      'Undefined Label-',0
0898' 69 6E 65 64 20
089D' 4C 61 62 65 6C
08A2' 2D 00
08A4' 53 74 61 63 6B   MSG7:DB      'Stack Overflow',0
08A9' 20 4F 76 65 72
08AE' 66 6C 6F 77 00
08B3'
08B3' 53 74 61 63 6B   MSG8:DB      'Stack Empty',0
08B8' 20 45 6D 70 74
08BD' 79 00
08BF' 49 6C 6C 65 67   MSG9:DB      'Illegal ORG',0
08C4' 61 6C 20 4F 52
08C9' 47 00
08CB'
08CB' D5               PRDEC:       PUSH    DE
08CC' F5                    PUSH    AF
08CD' AF                    XOR     A
08CE' 32 ** **         LD   (DECWK+5),A
08D1' 06 05                 LD      B,5
08D3' 11 ** **         LD   DE,DECWK+5
08D6' 0E 0A            PRDEC0:      LD  C,10
08D8' CD ** **         CALL DIVC
08DB' C6 30                 ADD     A,'0'
08DD' 1B                    DEC     DE
08DE' 12                    LD      (DE),A
08DF' 10 F5                 DJNZ    PRDEC0
08E1' 6B                    LD      L,E
08E2' 62                    LD      H,D
08E3' 06 04                 LD      B,4
08E5' 7E               PRDEC1:      LD  A,(HL)
08E6' FE 30                 CP      '0'
08E8' 20 05                 JR      NZ,PRDEC2
08EA' 36 20                 LD      (HL),' '
08EC' 23                    INC     HL
08ED' 18 F6                 JR      PRDEC1
08EF' CD E5 1F  PRDEC2: CALL _MSX
08F2' F1                    POP     AF
08F3' E1                    POP     HL
08F4' C9                    RET
08F5'                  ;
08F5'                  ; HL/C=HL...A
08F5'                  ;
08F5' C5               DIVC:PUSH    BC
08F6' AF                    XOR     A
08F7' 06 10                 LD      B,16
08F9' 29               DIVC0:       ADD HL,HL
08FA' 17                    RLA
08FB' 2C                    INC     L
08FC' 91                    SUB     C
08FD' 30 02                 JR      NC,DIVC1
08FF' 2D                    DEC     L
0900' 81                    ADD     A,C
0901' 10 F6            DIVC1:       DJNZ    DIVC0
0903' C1                    POP     BC
0904' C9                    RET
0905'
0905'
0905'                  ;
0905'                  ; Initialize Routine Part I
0905'                  ;
0905' AF               iniwk1::XOR  A
0906' 32 ** **         LD   (LOCFLG),A
0909' 32 ** **         LD   (UNDEF),A
090C' 32 ** **         LD   (FLGEX),A
090F' 21 00 02         LD   HL,200H
0912' 22 ** **         LD   (LBLPTR),HL
0915' 01 00 10         LD   BC,LBLMAX
0918' 21 00 60         LD   HL,LBLFLG
091B' 36 00            IWK1:LD      (HL),0
091D' 23                    INC     HL
091E' 0B                    DEC     BC
091F' 78                    LD      A,B
0920' B1                    OR      C
0921' 20 F8                 JR      NZ,IWK1
0923' CD ** **         CALL INIHSH       ;init hash table
0926' C9                    RET
0927'
0927'                  ;
0927'                  ; Initialize Routine part II
0927'                  ;
0927' AF               iniwk2::XOR  A
0928' 32 ** **         LD   (STLVL),A
092B' 32 ** **         LD   (WRPNT),A
092E' 3D                    DEC     A
092F' 32 ** **         LD   (RDPNT),A
0932' 21 00 00         LD   HL,0
0935' 22 ** **         LD   (NXTOFF),HL
0938' 22 ** **         LD   (LBLOFF),HL
093B' 22 ** **         LD   (PTRCD),HL
093E' 22 ** **         LD   (PTRDT),HL
0941' 22 ** **         LD   (PTRWK),HL
0944' 22 ** **         LD   (PTRFC),HL
0947' 22 ** **         LD   (CNTDT),HL
094A' C9                    RET
094B'                  ;
094B'                  ;    HL = HL * DE
094B'                  ;
094B' 44               MUL: LD      B,H
094C' 4D                    LD      C,L
094D' 21 00 00         LD   HL,0
0950' 3E 10                 LD      A,16
0952' 29               MUL0:ADD     HL,HL
0953' CB 23                 SLA     E
0955' CB 12                 RL      D
0957' 30 01                 JR      NC,MUL1
0959' 09                    ADD     HL,BC
095A' 3D               MUL1:DEC     A
095B' 20 F5                 JR      NZ,MUL0
095D' C9                    RET
095E'                  ;
095E'                  ;    HL/BC=HL...DE
095E'                  ;
095E' 42               DIV: LD      B,D
095F' 4B                    LD      C,E      ;LD BC,DE
0960' 54                    LD      D,H
0961' 5D                    LD      E,L      ;LD DE,HL
0962' 3E 10                 LD      A,16
0964' 26 00                 LD      H,0
0966' CB 23            DIV1:SLA     E
0968' CB 12                 RL      D
096A' 29                    ADD     HL,HL
096B' E5                    PUSH    HL
096C' B7                    OR      A
096D' ED 42                 SBC     HL,BC
096F' E1                    POP     HL
0970' 38 03                 JR      C,DIV2
0972' ED 42                 SBC     HL,BC
0974' 13                    INC     DE
0975'
0975' 3D               DIV2:DEC     A
0976' 20 EE                 JR      NZ,DIV1
0978' EB                    EX      DE,HL
0979' C9                    RET
097A'
097A'
097A'
097A'                  ; POKE with Increment HL
097A' CD 9A 1F  POKE_I: CALL _POKE
097D' 23                    INC     HL
097E' C9                    RET
097F'
097F'                  ; POKE BC
097F' F5               POKE_BC:PUSH AF
0980' 79                    LD      A,C
0981' CD 9A 1F         CALL _POKE
0984' 23                    INC     HL
0985' 78                    LD      A,B
0986' CD 9A 1F         CALL _POKE
0989' 23                    INC     HL
098A' F1                    POP     AF
098B' C9                    RET
098C'
098C'
098C'                  ; PEEK with Increment HL
098C' CD 94 1F  PEEK_I: CALL _PEEK
098F' 23                    INC     HL
0990' C9                    RET
0991'
0991'                  ; PEEK BC
0991' F5               PEEK_BC:PUSH AF
0992' CD 94 1F         CALL _PEEK
0995' 4F                    LD      C,A
0996' 23                    INC     HL
0997' CD 94 1F         CALL _PEEK
099A' 47                    LD      B,A
099B' 23                    INC     HL
099C' F1                    POP     AF
099D' C9                    RET
099E'
099E'                  ;
099E'                  ; Works
099E'                  ;
099E'                       DSEG
0000"
0000" 00 00 00 00 00   DECWK:       DEFW    0,0,0    ;work for PRDEC
0005" 00
0006"                  ITEM::       DS  1
0007"                  LOCFLG:      DS  1
0008"                  UNDEF:       DS  1
0009"                  SEGMD::      DS  1
000A"                  PTRFC::      DS  2
000C"                  PTRCD::      DS  2
000E"                  PTRDT::      DS  2
0010"                  PTRWK::      DS  2
0012"                  CNTCD::      DS  2        ;Counter of CSEG code
0014"                  CNTDT::      DS  2        ;Counter of DSEG Code
0016"                  FLGCD::      DS  1        ;Whether output CSEG
0017"                  FLGDT::      DS  1        ;Whether output DSEG
0018"                  FLGEX::      DS  1        ;Whether exec-address i
s designated
0019"
0019"                  EXADR::      DS  2        ;Executing Address
001B"
001B"                  SPBUF:       DS  2        ;SP buffer
001D"
001D"                  PRTLOC:      DS  1        ;Print Location
001E"                  PRTCNT:      DS  2        ;Counter used in PRTMAP
0020"
0020"                  STLVL:       DS  1        ;Stack Level
0021"                  STBUF:       DS  2 * STMAX    ;Stack Buffer
0031"
0031"                  LBLPTR:      DS  2
0033"                  LBLBUF::DS   32 + 1
0054"
0054"                  LBLOFF::DS   2        ;Local Label Number Offset
0056"                  NXTOFF::DS   2        ;LBLOFF for Next Module
0058"
0058"                       END
```

リスト4

```
0000'                  ;**********************************
0000'                  ; File Access Routine For WLK
0000'                  ;     Programed by T.Ishigami
0000'                  ;       '90 Feb.25th
0000'                  ;
0000'                  ;**********************************
0000'
0000'                       CSEG
0000'
0000'                C      INCLUDE SOS.DEF
1FFA                 C _HOT EQU     1FFAH
1FF4                 C _PRINT       EQU 1FF4H
1FF1                 C _PRNTS       EQU 1FF1H
1FEE                 C _LTNL        EQU 1FEEH
1FEB                 C _NL  EQU     1FEBH
1FE5                 C _MSX EQU     1FE5H
1FDF                 C _TAB EQU     1FDFH
1FD3                 C _GETL        EQU 1FD3H
1FC7                 C _PAUSE       EQU 1FC7H
1FC1                 C _PRTHX       EQU 1FC1H
1FBE                 C _PRTHL       EQU 1FBEH
1FB8                 C _HEX EQU     1FB8H
1FA3                 C _FILE        EQU 1FA3H
1FAF                 C _WOPEN       EQU 1FAFH
1F9A                 C _POKE        EQU 1F9AH
1F94                 C _PEEK        EQU 1F94H
2009                 C _ROPEN       EQU 2009H
2015                 C _KILL        EQU 2015H
2012                 C _NAME        EQU 2012H
2033                 C _ERROR       EQU 2033H
0000'                C
2000                 C _DRDSB       EQU 2000H
2003                 C _DWTSB       EQU 2003H
0000'                C
1F7A                 C _PRCNT       EQU 1F7AH
1F76                 C _KBFAD       EQU 1F76H
1F74                 C _IBFAD       EQU 1F74H
1F72                 C _SIZE        EQU 1F72H
1F6A                 C _MEMAX       EQU 1F6AH
1F64                 C _DTBUF       EQU 1F64H
1F62                 C _FATBF       EQU 1F62H
1F60                 C _DIRPS       EQU 1F60H
1F5E                 C _FATPOS      EQU 1F5EH
1F5D                 C _DSK EQU     1F5DH
0000'                C
0000'                C      INCLUDE WLK.DEF
0000'                C ; Header File For WLK
0000'                C ;    CSEG 3000H-
0000'                C ;    DSEG 4500H-
0000'                C ;
1000                 C LBLMAX       EQU 1000H
0000'                C
5000                 C cmdbuf       EQU 5000H    ;- 5FFFH
5F00                 C cmdlmt       EQU 5F00H
0000'                C
6000                 C LBLFLG       EQU 6000H    ;- 6FFFH
7000                 C LBLNUM       EQU 7000H    ;- 8FFFH
0000'                C
9000                 C BF_DSEG      EQU 9000H    ;- AFFFH
B000                 C RDBUF        EQU     0B000H   ;- B0FFH
B100                 C WRBUF        EQU     0B100H   ;- B1FFH
0000'
0000'
0000'                  ;====================
0000'                  ; FILE OPEN FOR READ
0000'                  ;====================
0000' CD A3 1F  RDOPEN::CALL _FILE
0003' D8                    RET     C
0004' CD ** **         CALL ROPEN
0007' D8                    RET     C
0008'                       ;
0008' 2A 74 1F         LD   HL,(_IBFAD)
000B' 11 ** **         LD   DE,FILE_BF
000E' 01 20 00         LD   BC,20H
0011' ED B0                 LDIR
0013' 3A 5D 1F         LD   A,(_DSK)
0016' 32 ** **         LD   (FLDSK),A
0019'
0019' AF                    XOR     A
001A' 32 ** **         LD   (LST_DSK),A ;NEVER BEING THE SAME
001D'
001D' CD ** **         CALL RDFAT
0020' D8                    RET     C
0021'
0021' 06 10                 LD      B,10H
0023' 0E 00                 LD      C,0      ; C <= TOTAL NUMBER OF CLUST
ERS
0025' 3A ** **         LD   A,(FSTCLST)
0028' 11 ** **         LD   DE,TBLCLST
002B'
002B' 12               RDOPN4:      LD  (DE),A
002C' 13                    INC     DE
002D' FE 7F                 CP      7FH
002F' 30 0F                 JR      NC,RDOPN5
0031' 2A 62 1F         LD   HL,(_FATBF)
0034' 85                    ADD     A,L
0035' 6F                    LD      L,A
0036' 30 01                 JR      NC,RDOPN2
0038' 24                    INC     H
0039' 7E               RDOPN2:      LD  A,(HL)       ;HL = (_FATBF) + (F
STCLST)
003A' 05                    DEC     B
003B' 28 28                 JR      Z,RDOPN3     ; More than 16 Cluster
003D' 0C                    INC     C
003E' 18 EB                 JR      RDOPN4
0040'
0040' 0D               RDOPN5:      DEC C        ;Last Cluster is Dummy
0041' F5                    PUSH    AF
0042' 79                    LD      A,C
0043' 87                    ADD     A,A
0044' 87                    ADD     A,A
0045' 87                    ADD     A,A
0046' 87                    ADD     A,A
0047' 4F                    LD      C,A      ;C = C * 16
0048' F1                    POP     AF
0049' D6 80                 SUB     80H      ;RC is 0 origin. So it isn't
 7Fh but 80H.
004B' 81                    ADD     A,C
004C' 32 ** **         LD   (RC),A      ;RC = Total number of Records
004F'
004F' AF                    XOR     A
0050' 32 ** **         LD   (FLPNT),A
0053'
0053' 01 37 00         LD   BC,IMF_SIZE
0056' 11 ** **         LD   DE,wkin
0059' 21 ** **         LD   HL,FILE_BF
```

▶予備校へ通いはじめて3カ月たったが，いつも，“また，来年ダメじゃないだろうなー”
と考えてしまう。助けてくれー！　もう狂気の世界はいやだー。　西野　由哲(19)愛知県

```
005C' ED B0                     LDIR
005E'
005E' 3E FF                     LD      A,0FFH
0060' 32 ** **          LD   (RDPNT),A   ;Clear Pointer For reading
0063'
0063' B7                        OR      A       ;CY = 0
0064' C9                        RET
0065'
0065' 3E 07             RDOPN3:         LD  A,7         ;Bad AllocationError
0067' 37                        SCF
0068' C9                        RET
0069'
0069'
0069' 3A 5D 1F  ROPEN:  LD   A,(_DSK)
006C' CD ** **          CALL DEVCHK
006F' D8                        RET     C       ;Bad File descripter
0070' CD ** **          CALL FCBSCH
0073' D8                        RET     C
0074' 3E 08                     LD      A,8     ;File Not Fonnd
0076' 37                        SCF
0077' C0                        RET     NZ
0078' E5                        PUSH    HL
0079' ED 5B 74 1F               LD      DE,(_IBFAD)
007D' 01 20 00          LD   BC,20H
0080' ED B0                     LDIR
0082' E1                        POP     HL
0083' 7E                        LD      A,(HL)
0084' CD ** **          CALL FMCHK
0087' C9                        RET
0088'
0088' FE 41             DEVCHK:         CP  'A'
008A' 38 04                     JR      C,DEVCH1
008C' FE 45                     CP      'D'+1
008E' 3F                        CCF
008F' D0                        RET     NC
0090'
0090' 3E 03             DEVCH1:         LD  A,3 ;Bad File descripter
0092' C9                        RET
0093'
0093'                   ;
0093'                   ; FCB SEARCH
0093'                   ;
0093' 0E 10             FCBSCH:         LD  C,16        ;Directory Lengh
0095' ED 5B 60 1F               LD      DE,(_DIRPS)  ;Directory start
0099' ED 53 ** **       FCBSC1:         LD  (DEBUF),DE
009D' 2A 64 1F          LD   HL,(_DTBUF)
00A0' 3E 01                     LD      A,1
00A2' CD 00 20          CALL _DRDSB
00A5' D8                        RET     C
00A6' 06 08                     LD      B,8
00A8' 22 ** **  FCBSC2: LD   (HLBUF),HL
00AB' 7E                        LD      A,(HL)
00AC' FE FF                     CP      0FFH
00AE' 28 1A                     JR      Z,FCBSC4
00B0' B7                        OR      A
00B1' 28 0B                     JR      Z,FCBSC3
00B3' D5                        PUSH    DE
00B4' ED 5B 74 1F               LD      DE,(_IBFAD)
00B8' CD ** **          CALL FCOMP
00BB' D1                        POP     DE
00BC' 28 0D                     JR      Z,FCBSC5
00BE' D5                FCBSC3:         PUSH    DE
00BF' 11 20 00          LD   DE,32
00C2' 19                        ADD     HL,DE
00C3' D1                        POP     DE
00C4' 10 E2                     DJNZ    FCBSC2
00C6' 13                        INC     DE
00C7' 0D                        DEC     C
00C8' 20 CF                     JR      NZ,FCBSC1
00CA'
00CA' 3E                FCBSC4:         DB  3EH ; Z = 0
00CB' AF                FCBSC5:         XOR A   ; Z = 1
00CC' B7                        OR      A
00CD' C9                        RET
00CE'
00CE'
00CE'                   ; File Name Compare
00CE' C5                FCOMP:          PUSH    BC
00CF' E5                        PUSH    HL
00D0' 06 10                     LD      B,16    ;Directory lengh
00D2' 13                FCOMP1:         INC DE
00D3' 23                        INC     HL
00D4' 1A                        LD      A,(DE)
00D5' BE                        CP      (HL)
00D6' 20 02                     JR      NZ,FCOMP2
00D8' 10 F8                     DJNZ    FCOMP1
00DA' E1                FCOMP2:         POP HL
00DB' C1                        POP     BC
00DC' C9                        RET
00DD'
00DD'                   ; FILE MODE CHECK
00DD' E5                FMCHK:          PUSH    HL
00DE' E6 87                     AND     87H     ;1000&0111B
00E0' 21 1F 29          LD   HL,291FH    ;%FTYPE
00E3' BE                        CP      (HL)
00E4' E1                        POP     HL
00E5' C8                        RET     Z
00E6' 3E 06                     LD      A,6     ;Bad File Mode
00E8' 37                        SCF
00E9' C9                        RET
00EA'
00EA'
00EA'                   ;====================
00EA'                   ; FILE OPEN FOR  WRITE
00EA'                   ;====================
00EA' CD A3 1F  WROPEN::CALL _FILE
00ED' D8                        RET     C
00EE' CD AF 1F          CALL _WOPEN
00F1' D8                        RET     C
00F2'                           ;
00F2' 01 20 00          LD   BC,20H
00F5' 11 ** **          LD   DE,FILE_BF
00F8' 2A 74 1F          LD   HL,(_IBFAD)
00FB' ED B0                     LDIR
00FD'
00FD' 3A 5D 1F          LD   A,(_DSK)
0100' 32 ** **          LD   (FLDSK),A
0103'
0103' 2A E1 27          LD   HL,(27E1H)
0106' 22 ** **          LD   (HLBUF),HL
0109' 2A DF 27          LD   HL,(27DFH)
010C' 22 ** **          LD   (DEBUF),HL
010F'
010F'                           ;CALL   RDFAT           ;FAT has already loaded
in _WOPEN
010F'
010F' CD ** **          CALL FCGET
0112' D8                        RET     C
0113'
0113' 32 ** **          LD   (FSTCLST),A
0116' 32 ** **          LD   (TBLCLST),A
0119' 3E 80                     LD      A,80H
011B' 32 ** **          LD   (TBLCLST+1),A
011E'
011E' AF                        XOR     A
011F' 32 ** **          LD   (RC),A
0122' 32 ** **          LD   (LST_DSK),A ;Never beibg then same
0125' 32 ** **          LD   (FLPNT),A
0128'
0128' 21 00 00          LD   HL,0
012B' 22 ** **          LD   (FLSIZE),HL
012E'
012E' 01 37 00          LD   BC,IMF_SIZE
0131' 11 ** **          LD   DE,wkout
0134' 21 ** **          LD   HL,FILE_BF
0137' ED B0                     LDIR
0139'
0139' 3E 00                     LD      A,0
013B' 32 ** **          LD   (WRPNT),A
013E'
013E' B7                        OR      A       ;RCF
013F' C9                        RET
0140'
0140'                   ;==============
0140'                   ; FILE CLOSE
0140'                   ;==============
0140' 21 ** **  CLOSE:: LD   HL,wkout
0143' 11 ** **          LD   DE,FILE_BF
0146' 01 37 00          LD   BC,IMF_SIZE
0149' ED B0                     LDIR
014B'
014B' 3A ** **          LD   A,(FLDSK)
014E' 32 5D 1F          LD   (_DSK),A
0151'
0151' 2A ** **          LD   HL,(FLSIZE)
0154' 2C                        INC     L
0155' 2D                        DEC     L
0156' C4 ** **          CALL NZ,WRITE1
0159'
0159' 3A ** **          LD   A,(RC)
015C' 2A ** **          LD   HL,(FLSIZE)
015F' 2C                        INC     L
0160' 2D                        DEC     L
0161' 20 01                     JR      NZ,COL3
0163' 3C                        INC     A
0164' 67                COL3:LD         H,A
0165' 22 ** **          LD   (FLSIZE),HL
0168'
0168' 3E 01                     LD      A,1
016A' ED 5B ** **               LD      DE,(DEBUF)
016E' 2A 64 1F          LD   HL,(_DTBUF)
0171' CD 00 20          CALL _DRDSB
0174' D8                        RET     C
0175'
0175' 21 ** **          LD   HL,FILE_BF
0178' ED 5B ** **               LD      DE,(HLBUF)
017C' 01 20 00          LD   BC,20H
017F' ED B0                     LDIR
0181'
0181' 3E 01                     LD      A,1
0183' ED 5B ** **               LD      DE,(DEBUF)
0187' 2A 64 1F          LD   HL,(_DTBUF)
018A' CD 03 20          CALL _DWTSB
018D' D8                        RET     C
018E'
018E' CD ** **          CALL RDFAT
0191' D8                        RET     C
0192'
0192' 06 10                     LD      B,10H
0194' 21 ** **          LD   HL,TBLCLST
0197' 7E                COL1:LD         A,(HL)
0198' FE 7F                     CP      7FH
019A' 30 12                     JR      NC,COL2
019C'
019C' 23                        INC     HL
019D' 4E                        LD      C,(HL)
019E' E5                        PUSH    HL
019F' 2A 62 1F          LD   HL,(_FATBF)
01A2' 16 00                     LD      D,0
01A4' 5F                        LD      E,A
01A5' 19                        ADD     HL,DE
01A6' 71                        LD      (HL),C
01A7' E1                        POP     HL
01A8'
01A8' 05                        DEC     B
01A9' CA ** **          JP   Z,RDOPN3    ;Bad File Allocation
01AC' 18 E9                     JR      COL1
01AE'
01AE' CD ** **  COL2:   CALL WRFAT
01B1' C9                        RET
01B2'
01B2'                   ;=================
01B2'                   ; INPUT      FROM FILE
01B2'                   ;=================
01B2'                   INPUT_::
01B2' 21 ** **          LD   HL,RDPNT
01B5' 34                        INC     (HL)
01B6' 20 08                     JR      NZ,INP1      ;INC (RDPNT) lower byte
01B8'
01B8' CD ** **          CALL READ
01BB' D8                        RET     C
01BC' 21 ** **          LD   HL,wkin + 31H   ;HL points FLPNT
01BF' 34                        INC     (HL)
01C0'
01C0' 2A ** **  INP1:   LD   HL,(RDPNT)
01C3' 7E                        LD      A,(HL)
01C4' B7                        OR      A       ;RCF
01C5' C9                        RET
01C6'
01C6'
01C6'                   ;====================
01C6'                   ; Seek for input file
01C6'                   ;====================
01C6' 7D                FSEEK::         LD  A,L
01C7' 3D                        DEC     A
01C8' 32 ** **          LD   (RDPNT),A
01CB' 7C                        LD      A,H
01CC' 32 ** **          LD   (wkin + 31H),A  ;LD (FLPNT),A
01CF' CD ** **          CALL READ
01D2' D8                        RET     C
01D3' 21 ** **          LD   HL,wkin + 31H
01D6' 34                        INC     (HL)
01D7' B7                        OR      A       ;CY = 0
01D8' C9                        RET
01D9'
01D9' 00 B0             RDPNT::         DW  RDBUF
01DB'
01DB'
01DB' 01 37 00  READ:   LD   BC,IMF_SIZE
01DE' 11 ** **          LD   DE,FILE_BF
01E1' 21 ** **          LD   HL,wkin
01E4' ED B0                     LDIR
01E6'
01E6' 3A ** **          LD   A,(FLDSK)
01E9' 32 5D 1F          LD   (_DSK),A
01EC'
01EC' 3A ** **          LD   A,(FLPNT)
01EF' 47                        LD      B,A
01F0' 3A ** **          LD   A,(RC)
01F3' B8                        CP      B
01F4' DA ** **          JP   C,RDOPN3    ;Bad AllocationFile
01F7'
01F7' 78                        LD      A,B
01F8' CD ** **          CALL PNTREC
01FB' EB                        EX      DE,HL         ; DE = RECORD
01FC'
01FC' 3E 01                     LD      A,1
01FE' 21 00 B0          LD   HL,RDBUF
0201' CD 00 20          CALL _DRDSB
0204' C9                        RET
0205'
0205'                   ;==============
0205'                   ; PRINT      TO FILE
0205'                   ;==============
0205' 2A ** **  PRINT_::LD   HL,(wkout+12H)
0208' 23                        INC     HL
0209' 22 ** **          LD   (wkout+12H),HL  ;(FLSIZE)++
020C'
020C' 2A ** **          LD   HL,(WRPNT)
020F' 77                        LD      (HL),A
0210'
0210' 21 ** **          LD   HL,WRPNT
0213' 34                        INC     (HL)          ;INC (WRPNT1) lower byt

0214' 37                        SCF
0215' 3F                        CCF             ;CY = 0
0216' C0                        RET     NZ
0217'
0217' CD ** **          CALL WRITE
021A' D8                        RET     C
021B'
021B' 21 ** **          LD   HL,wkout+31H    ;HL points FLPNT
021E' 34                        INC     (HL)
021F' B7                        OR      A       ;RCF
0220' C9                        RET
0221'
0221' 00 B1             WRPNT::DW       WRBUF
0223'
0223'
0223' 01 37 00  WRITE:  LD   BC,IMF_SIZE
0226' 11 ** **          LD   DE,FILE_BF
0229' 21 ** **          LD   HL,wkout
022C' ED B0                     LDIR
022E'
022E' 3A ** **          LD   A,(FLDSK)
0231' 32 5D 1F          LD   (_DSK),A
0234'
0234' 3A ** **  WRITE1: LD   A,(FLPNT)
0237' 47                        LD      B,A
0238' 3A ** **          LD   A,(RC)
023B' B8                        CP      B
023C' 30 76                     JR      NC,WRITE4
023E'
023E' CD ** **          CALL RDFAT
0241' D8                        RET     C
0242'
0242' 3A ** **          LD   A,(FLPNT)
0245' E6 F0                     AND     0F0H
0247' 47                        LD      B,A
0248' 3A ** **          LD   A,(RC)
024B' E6 F0                     AND     0F0H
024D' B8                        CP      B
024E' 30 47                     JR      NC,WRITE2
0250'
0250' CB 3F                     SRL     A
0252' CB 3F                     SRL     A
0254' CB 3F                     SRL     A
0256' CB 3F                     SRL     A       ;A = A / 16
0258' 21 ** **          LD   HL,TBLCLST+1
025B' 16 00                     LD      D,0
025D' 5F                        LD      E,A
025E' 19                        ADD     HL,DE
025F' E5                        PUSH    HL
0260'
0260' 3A ** **          LD   A,(RC)
0263' E6 F0                     AND     0F0H
0265' CD ** **          CALL PNTREC
0268' CD ** **          CALL RECCL
026B'
026B' 2A 62 1F          LD   HL,(_FATBF)
026E' 16 00                     LD      D,0
0270' 5F                        LD      E,A     ;A = RC && 0F0H
0271' 19                        ADD     HL,DE
0272' 36 8F                     LD      (HL),8FH        ;DUMMY
0274' CD ** **          CALL FCGET
0277' 77                        LD      (HL),A
0278' E1                        POP     HL
0279' D8                        RET     C
027A' 77                        LD      (HL),A
027B' 23                        INC     HL
027C' 36 80                     LD      (HL),80H        ;Debugged '90 Mar 8th
027E'
027E' 2A 62 1F          LD   HL,(_FATBF)
0281' 16 00                     LD      D,0
0283' 5F                        LD      E,A
0284' 19                        ADD     HL,DE
0285' 36 80                     LD      (HL),80H
0287' 3A ** **          LD   A,(RC)
028A' E6 F0                     AND     0F0H
028C' C6 10                     ADD     A,16            ;ONE CLUSTER ADDED
028E' 32 ** **          LD   (RC),A
0291'
0291' CD ** **          CALL   WRFAT
0294' D8                        RET    C
0295' 18 9D                     JR      WRITE1
0297'
0297' 3A ** **  WRITE2: LD   A,(FLPNT)
029A' 32 ** **          LD   (RC),A
029D' CB 3F                     SRL     A
029F' CB 3F                     SRL     A
02A1' CB 3F                     SRL     A
02A3' CB 3F                     SRL     A       ;A = A / 16
02A5' 21 ** **          LD   HL,TBLCLST+1
02A8' 16 00                     LD      D,0
02AA' 5F                        LD      E,A
02AB' 19                        ADD     HL,DE
02AC'
02AC' 3A ** **          LD   A,(FLPNT)
02AF' E6 0F                     AND     0FH
02B1' C6 80                     ADD     A,80H
02B3' 77                        LD      (HL),A
02B4'
02B4' 3A ** **  WRITE4: LD   A,(FLPNT)
02B7' CD ** **          CALL PNTREC
02BA' EB                        EX      DE,HL           ;DE = RECORD NO.
02BB'
02BB' 3E 01                     LD      A,1
02BD' 21 00 B1          LD   HL,WRBUF
02C0' CD 03 20          CALL _DWTSB
02C3' D8                        RET     C
02C4'
02C4' 01 37 00          LD   BC,IMF_SIZE
02C7' 11 ** **          LD   DE,wkout
02CA' 21 ** **          LD   HL,FILE_BF
02CD' ED B0                     LDIR
02CF' B7                        OR      A       ;RCF
02D0' C9                        RET
02D1'
02D1'
02D1'                   ; FILE POINTER (A) => RECORD NO. (HL)
02D1' F5                PNTREC:         PUSH    AF
02D2' F5                        PUSH    AF
02D3' CB 3F                     SRL     A
02D5' CB 3F                     SRL     A
02D7' CB 3F                     SRL     A
02D9' CB 3F                     SRL     A       ;A = A / 16
02DB' 21 ** **          LD   HL,TBLCLST
02DE' 16 00                     LD      D,0
02E0' 5F                        LD      E,A
02E1' 19                        ADD     HL,DE
02E2' 7E                        LD      A,(HL)
02E3' CD ** **          CALL CLREC
02E6' F1                        POP     AF
02E7' E6 0F                     AND     0FH
02E9' 85                        ADD     A,L
02EA' 6F                        LD      L,A
02EB' F1                        POP     AF
02EC' C9                        RET
02ED'
02ED'
02ED'                   ; FAT READ TO BUFFER
02ED' D5                RDFAT:          PUSH    DE
02EE' E5                        PUSH    HL
02EF' 3A ** **          LD   A,(LST_DSK)
02F2' 57                        LD      D,A
02F3' 3A 5D 1F          LD   A,(_DSK)
02F6' BA                        CP      D
02F7' 28 0F                     JR      Z,RDFAT_1
02F9' 32 ** **          LD   (LST_DSK),A
02FC' 3E 01                     LD      A,1
02FE' ED 5B 5E 1F               LD      DE,(_FATPOS)
0302' 2A 62 1F          LD   HL,(_FATBF)
0305' CD 00 20          CALL _DRDSB
0308' E1                RDFAT_1:POP  HL
0309' D1                        POP     DE
030A' C9                        RET
030B'
030B'                   ; FAT WRITE FROM BUFFER
030B' D5                WRFAT:          PUSH    DE
030C' E5                        PUSH    HL
030D' 3E 01                     LD      A,1
030F' ED 5B 5E 1F               LD      DE,(_FATPOS)
0313' 2A 62 1F          LD   HL,(_FATBF)
0316' CD 03 20          CALL _DWTSB
0319' E1                        POP     HL
031A' D1                        POP     DE
031B' C9                        RET
031C'
031C'                   ; FREE CLUSTER POSITION   GET
031C' C5                FCGET:          PUSH    BC
031D' E5                        PUSH    HL
031E' 06 80                     LD      B,80H
0320' 2A 62 1F          LD   HL,(_FATBF)
0323' 7E                FCGET2:         LD  A,(HL)
0324' B7                        OR      A
0325' 28 08                     JR      Z,FCGET3
0327' 23                        INC     HL
0328' 10 F9                     DJNZ    FCGET2
032A' 3E 09                     LD      A,9     ;Device  Full
032C' 37                        SCF
032D' 18 04                     JR      FCGET4
032F' 3E 80             FCGET3:         LD  A,80H
0331' 90                        SUB     B
0332' A7                        AND     A
0333' E1                FCGET4:         POP HL
0334' C1                        POP     BC
0335' C9                        RET
0336'
0336'                   ; CLUSTER (A) => RECORD   (HL)
0336' 26 00             CLREC:          LD  H,0
0338' 6F                        LD      L,A
0339' 29                        ADD     HL,HL
033A' 29                        ADD     HL,HL
033B' 29                        ADD     HL,HL
033C' 29                        ADD     HL,HL
033D' C9                        RET
033E'
033E'                   ; RECORD (HL) => CLUSTER (A)
033E' E5                RECCL:          PUSH    HL
```


▶編集部の住所が変わってしまった。せっかく全日本プロレス武道館大会の帰りに編集室に結果報告に行こうと思っていたのに。高輪かぁ。慶応か明治学院大学でも受験しようかなあ。

津田　典秀(20)千葉県

```
033F' CB 3C          SRL    H
0341' CB 1D          RR     L   ;HL/2
0343' CB 3C          SRL    H
0345' CB 1D          RR     L   ;HL/4
0347' CB 3C          SRL    H
0349' CB 1D          RR     L   ;HL/8
034B' CB 3C          SRL    H
034D' CB 1D          RR     L   ;HL/16
034F' 7D             LD     A,L
0350' E1             POP    HL
0351' C9             RET
0352'
0352'       ;
0352'       ; Works
0352'       ;
0352'
0352'                DSEG
0000"
0000"       FCB_ADR:DS   2      ;File Control Block Address
0002"       LST_DSK:DS   1      ;Disk drived last
0003"
0003"
0037        IMF_SIZE    EQU 37H
0003"       wkin::      DS  IMF_SIZE ;work area for Input-fi
le
003A"       wkout::     DS  IMF_SIZE ;work area for Output-f
ile
0071"
0071"       FILE_BF:DS  12H
0083"       FLSIZE:     DS  2   ;File Size.
0085"                          FLDTADR:DS   2   ;Start Address.
0087"                          FLEXADR:DS   2   ;Exec Address.
0089"                               DS      6
008F"                          FSTCLST:DS   1   ;First Cluster.
0090"                               DS      1
0091"                          TBLCLST:DS   10H ;Cluster table.
00A1"                          FLDSK:       DS 1   ;The Login disk.
00A2"                          FLPNT:       DS 1   ;The FILE Pointor.
00A3"                          DEBUF:       DS 2   ;Record No.Which Have The DI
R.
00A5"                          HLBUF:       DS 2   ;Address Where On IMFORMATIO
N.
00A7"                          RC:  DS      1   ;The Number  of Records The File
 have.
00A8"
00A8"                               END
```

リスト5

```
; Header File For WLK
;   CSEG 3000H-
;   DSEG 4500H-
;
LBLMAX          EQU        1000H

cmdbuf          EQU       5000H      ;- 5FFFH
cmdlmt          EQU       5F00H

LBLFLG          EQU       6000H      ;- 6FFFH
LBLNUM          EQU       7000H      ;- 8FFFH

BF_DSEG         EQU       9000H      ;- AFFFH
RDBUF           EQU       0B000H     ;- B0FFH
WRBUF           EQU       0B100H     ;- B1FFH
```

リスト6

```
; Header File For WZD
;   CSEG 3000H-
;   DSEG 6000H-
;

RDBUF           EQU 0B800H
WRBUF1          EQU 0BC00H
WRBUF2          EQU 0BD00H

```

全機種共通システムインデックス

■85年6月号
序論　共通化の試み
第1部　S-OS"MACE"
第2部　Lisp-85インタプリタ
第3部　チェックサムプログラム
■85年7月号
第4部　マシン語プログラム開発入門
第5部　エディタアセンブラZEDA
第6部　デバッグツールZAID
■85年8月号
第7部　ゲーム開発パッケージBEMS
第8部　ソースジェネレータZING
■85年9月号
インタラプト　S-OS番外地
第9部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表
■86年1月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年2月号
第15部　S-OS"SWORD"
第16部　Prolog-85入門(1)
■86年3月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年4月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年5月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年6月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ&エディタ
第24部　"SWORD" 2000 QD
連載　対話で学ぶ magiFORTH
特別付録　PC-8801版S-OS"SWORD"
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版 S-OS"SWORD"
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版 S-OS"SWORD"
■86年9月号
第28部　FuzzyBASIC 発表
連載　明日に向かって magiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタ DREAM
第31部　FuzzyBASIC 料理法〈1〉
■86年11月号
第32部　パズルゲーム HOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC 料理法〈2〉
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC 料理法〈3〉
■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC 料理法〈4〉
■87年2月号
第36部　アドベンチャーゲーム MARMALADE
第37部　テキアペ作成ツール CONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツール MAGE
付録　"SWORD" 再掲載と MAGIC の標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS"SWORD" 変身セット
第43部　MZ-700用 "SWORD" を QD 対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASIC コンパイラ
第45部　エディタアセンブラ ZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージ JACKWRITE
特別付録　FM-7/77版 S-OS"SWORD"
■87年9月号
第49部　リロケータブル逆アセンブラ Inside-R
特別付録　PC-8001/8801 版 S-OS"SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASIC コンパイラの拡張
第52部　X1turbo 版 S-OS"SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OS の仲間たち
第53部　もうひとつの FuzzyBASIC 入門
第54部　ファイルアロケータ&ローダ
インタラプト　S-OS こちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo 版 "SWORD" アフターケア
　　ラインプリントルーチン
特別付録　PASOPIA7 版 S-OS"SWORD"
■88年1月号
第58部　FuzzyBASIC コンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲーム ELFES
■88年3月号
第60部　構造型コンパイラ言語 SLANG
■88年4月号
第61部　デバッギングツール TRADE
第62部　シミュレーションウォーゲーム WALRUS
■88年5月号
第63部　シューティングゲーム ELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語 SLANG 入門(1)
第66部　Lisp-85 用 NAMPA シミュレーション
■88年7月号
第67部　マルチウィンドウドライバ MW-1
連載　構造化言語 SLANG 入門(2)
■88年8月号
第68部　マルチウィンドウエディタ WINER
■88年9月号
第69部　超小型エディタ TED-750
第70部　アフターケア WINER の拡張
■88年10月号
第71部　SLANG 用ファイル入出力ライブラリ
第72部　シューティングゲーム MANKAI
■88年11月号
第73部　シューティングゲーム ELFES Ⅳ
■88年12月号
第74部　ソースジェネレータ SOURCERY
■89年1月号
第75部　パズルゲーム LAST ONE
第76部　ブロックゲーム FLICK
■89年2月号
第77部　高速エディタアセンブラ REDA
特別付録　X1版 S-OS"SWORD"〈再掲載〉
■89年3月号
第78部　Z80用浮動小数点演算パッケージSOROBAN
■89年4月号
第79部　SLANG 用実数演算ライブラリ
■89年5月号
第80部　ソースジェネレータ RING
■89年6月号
第81部　超小型コンパイラTTC
■89年7月号
第82部　TTC用パズルゲーム　TICBAN
■89年8月号
第83部　CP/M用ファイルコンバータ
■89年9月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲーム PUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリ
　　DIO. LIB
■90年1月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年2月号
第89部　超小型コンパイラTTC++
■90年3月号
第90部　超多機能アセンブラOHM-Z80
■90年4月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部　インタプリタ言語STACK
■90年6月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲーム SQUASH！
第95部　X68000対応S-OS"SWORD"
特別付録　PC-286対応S-OS"SWORD"
■90年7月号
第96部　リロケータブルアセンブラWZD

＊以上のアプリケーションは，基本システムである S-OS "MACE" または S-OS"SWORD" がないと動作しませんのでご注意ください。

THE SENTINEL

▶7月号は何といっても133ページの予告，西川さんのFAXの絵が一番面白かった。「勝てるものなら勝ってみろ」。んー，相当自信があるんだなあ～。結果が楽しみだ。しかし，「あらかじめご了承ください」攻撃はやめなさい!?　功刀　和久(21)埼玉県

人工知能の冒険

完全な真空

毛色の変わった本を出すとして有名な国書刊行会から出されている『完全な真空』という，おかしな本をぜひとも皆さんに紹介しようと思います。著者はスタニスワフ・レムでして，タルコフスキーの撮った「惑星ソラリス」という映画の原作者として有名です。レムはSF作家としてきわめて有名であり,「最高のSF作家」とさえ呼ばれているそうです。

この『完全な真空』は，本当はこの世に存在しない本を，まるで存在するかのように出版社や作者名まででっち上げたうえ，それらの本それぞれに対する書評をまた自分で書いているというものです。ひとことでいえば，架空書評集というところでしょう。

全部で16冊の架空の書物が取り上げられているのですが，おもしろいものとおもしろくないものの差がきわめて大きく感じられました。あまり興味がもてなかったのが,『親衛隊少将ルイ16世』や『白痴』のように，なにかスケールの異常に大きい大作の概略を示したようなものです。逆に，3度も4度も読んでしまったのが,「最高のSF作家」こそが書きうるというようなものです。

実在する1冊の本

正確にいうならば，この本に収録されている16冊の架空書物のうち，先頭に取り上げられている1冊だけは実在します。それは，この本『完全な真空』自体です。そこでは，まるで別の人が書いたように，「レム氏は……」などとしらばっくれて書いています。しかもさらに，その文章の中で，その文章そのものを引用することまで行い，一層混乱の度をわざと高めています。

書き上げてもいない本を作り出し，それを今度は評論家の立場で好き勝手に批評し，そうしてできた本をまた同じ本の中で批評するとは，いってみればなんともの書きとしてぜいたくなことをやっているのだろうと思わず感じてしまいます。

・このように書評集の中でその書評集自体を取り上げるというのは，「再帰呼び出し」(リカーシブコール）を思い起こさせます。この例に見られるように，再帰呼び出し的なことは単にプログラムの中の関数の呼び出し方だけに限定された話ではありません。ネーミングの中に見られるごく簡単な例を示しましょう。UNIXオペレーティングシステムの発展版にそのスペルを引っ繰り返したXINU（ジーニュ）というのがありますが，これは次の文章の頭文字をとったものだそうです。

“XINU Is Not Unix.”

研究室にあるUNIXマシンのひとつ（CPU はSPARC）の名前を，SPARCを引っ繰り返したCRAPSとしているのですが，その名前の由来も無理やりこのXINUのように説明するならば，

“CRAPS Runs A Processor Sparc.”
(CRAPSはSparcプロセッサを駆動する）とでもいえばよいのでしょう。

存在しえない小説

『完全な真空』の中で取り上げられている仮想小説のうちのひとつに「とどのつまりは何も無し」というものがあります。この小説についてここで紹介し，読者の皆さんにああこういう小説なのかとわかってもらうことは，きわめてむずかしいことと思われます。第一，僕自身どう考えても，このような小説がどのように存在し得るか，あまり想像がつかないからです。

まあとにかく，この小説の内容を紹介することにしましょう（無駄とわかっていても)。この小説の内容はないのです。といっても，真っ白な紙が並んでいるのではなく，しっかりと文章が並んでいるのです(もちろん,「何もない」と1000回書かれているわけでもありません)。しかし，何も語ってはいないのです。

冒頭の文は「列車は着かなかった」となっています。そして,「誰か」が現れなかったあと，語りは非人称のまま，時は春でもなく夏でもなく，無重力空間における愛されない女に関する考察によって第1章は閉じられます。

その後，この本に関する記述は抽象度を増します。「虚無の穴が不気味に大きくなってゆく」「思考しないことの流れ」「テキストはわれわれの所有していたものを次々と奪い取っていく」……。作品の最後ではもうこれ以上作品が続きうるかという疑念が沸き起こってきます。

そして，ついには「存在しないこと」は否定として存在することさえやめてしまうのです。文章の意味が失われると残るのは構文のみです。しかしその文法装置さえしまいには空中分解してしまい，文章の途中，単語の途中でついにこの小説は終わってしまうのです（とまあちょっとだけ書いてみましたが，やはり徒労に終わったのでしょうね？)。

でも，実際には存在しえない小説を仮想することこそ，この本の真価といえるでしょう。しかもなぜこのような小説がこの世に存在するかという意味づけもしっかりとなされています。要するに，小説家が誠実さを究極にまで追求したときに必然的に生まれる小説は，まさにこのようなものであるということです。小説家はありもしないことを書かなくてはならないのですが，もしそのような行為に良心の呵責を感じるような小説家が万一存在したならば，彼の取るべき道は2つだけ，筆を折るか，あるいは「とどのつまりは何も無い」小説を書くかということなのです。

このような小説を書く小説家の誠実さについて論じながらレムは，「私はそのような意味での誠実さからはいちばん遠いのだ」と含み笑いしていることでしょう，小説家が誠実さを求めることは，レムの行っている「ありもしない小説をでっち上げる」行為とちょうど正反対であるからなのです。

ところで，この世に存在しない小説の書評をした本を取り上げて，それをまた書評している僕自身の誠実さはいったいどうな

っているのでしょうか？　まあ，この『完全な真空』という本が存在しないのならば，それこそ賞賛に値するほどの不誠実さとでもいえるでしょうが，僕はまだまだ……。

知能の相対化

既成のとらわれた概念に対する鋭い風刺の効いた疑問は，この本のいろいろなところに見られます。「誤謬としての文化」では，まず，「文化は生物が生き残る邪魔にもならなければ，助けにもならないものである」という考えを否定します（これはまあ普通の主張といえましょう）。ところが次に主張されるのはきわめて刺激的な考えです。「文化というものは，自ら作り出した宗教，慣習，法，禁止，命令を通じて作用することにより，不十分なものを理想に，マイナスをプラスに，欠点や欠陥のあるものを完璧なものに作り変えるのだ」というのです。

あるいは別の書評では，知能というものに関して，人間の知能の絶対性というものに強い懐疑を示します。そしてこれは，「完全な真空」以外の彼の書物にも見られる一貫した態度のようです。人工知能という言葉は，最近ではごく当たり前に使われる言葉になってきたのですが，その際，知能は人間の頭脳こそが唯一もっているものであるということは，当然のこと，暗黙の了解事項であるように僕には感じられます。「ソラリス」のテーマ自体がそうであったように，レムはいつも人間のもっているものが知能として絶対唯一であるということへの疑問を提示しています。それどころか，この本を読むと人間の知能などは偶然の産物なのだという声さえ，きわめて皮肉的かつ間接的ではありますが，聞こえてきます。

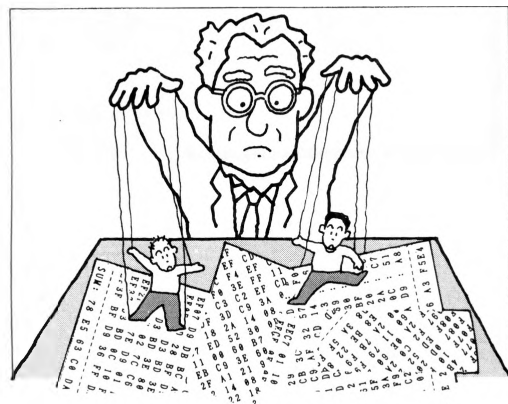
この本が書かれたのはなんと1971年です（日本語訳が出たのは1989年）。その後10年くらいたって，いわゆるサイバーパンクといわれる新しい潮流が生まれて，人間の脳の神経細胞のクローズアップ，たとえば，直接，神経細胞をメディアとしてコミュニケーションするという考えなどが生まれたわけです。次に紹介する架空書物評などを読むと，この本が今から20年も前に書かれたとは信じられない気がします。

「我は下僕ならずや」では，現実世界からはまったく切り離された神経細胞における電気パルスの伝達でのみ構成される世界というものを，さらに独立させ，純粋化した世界を描いています。キーワードはパーソネティクス（理性ある生物の人工生産）なのだそうです。そのような世界を小説として描いているのではなく，実際にそのような世界を研究室の計算機内に作り上げたドブ教授がこの架空小説の著者なのです。

もうひとつ別の人工知能

「我は下僕ならずや」で記述されている世界といっても，完全に計算機の中の閉じた世界であり，まったく数学的に作り上げられたものなのです。しかしそこには「住人」がいるのです。なぜそこに，知能をもつ生命体が住んでいるとみなせるかというところがミソであり，えんえんとページが費やされています。いわゆるシミュレーションのようにも思えるのですが，シミュレーションではありません，実体なのです。現実の物理的空間がないところになぜ知的生命体が想定できるのでしょうか？

このことについては，人間が住んでいるこの世というものが実は偶然の産物であり，数学的な世界の中にも，人間世界とまったく同じような現象が起こりうるということを執拗に述べています。偶然この世は3次元なのですが，彼ら「住人」の住む数学的世界ではそれらが任意に（ドブ教授によって）設定することができるというのです。時間の進み具合も設定できます。ある種の具体化を遂げた数学は，完全に実体をもたぬほどに精神化した知性の生活空間となりえたのです。さらにレムは，この世や人間に特有なさまざまな概念，たとえば，意識，言語，進化などに関して，その脆弱性（もろくて弱いということ）を追求し，そして計算機の中に閉じ込められた世界でも同様の概念が存在するというのです。



この架空小説が最大に盛り上がるのは，「住人」たちの，創造主（つまりこの架空小説を書いているドブ教授）に関する議論です。何人かの「住人」たちが，いったい創造主はいるのかいないのか，いるのならば，今の我々とどういう関係にあるのかということを話し合うのです。おもしろいのは，彼らの世界を述べているようで，いつの間にか，実は我々人間自身の問題と完全にオーバーラップしてくることです。

レムは実は計算機の中の人間が作り出した世界に生きる知的生命体を描きながら，実は，我々もまた上のレベルにある何者か（創造主）に操られているというような循環をも同時に描いているのでしょう。

知能機械といっても，人がもっているような知能だけを相手にしているのではもう古いのかもしれません。50年先，500年先をにらんで生きていく人は，ソラリスの海やサイバーパンクやパーソネティクスまでをも包括したものとして，知能というものをイメージしていかねばならないのでしょう。

というわけで，本連載でも，総力を込めてというか，脱線しまくってというか，次回には，毛色の全然違う未知の領域に踏み込もうと思います。タイトルは，「超能力大実験：ここにも超能力者が！（仮称）」です。（こりゃとんだことになりそうだと感じつつ）来月をお楽しみに！

第50回

猫とコンピュータ

サーチャーでござる

Takazawa Kyoko
高沢 恭子



あれってどこ置いたんだっけ？　っていうときは，自分がそれを置きそうなところや隠れそうなところをさがしますよね。ホンニャアにしても同じこと，長年培われた体験がさがしものにはモノをいうようです。

ホンニャアは体内に上等のセンサーがあるから，日に日に近づく灼熱の季節を，もう感じている。うすぐもりの空と湿った風にくるまれて，太陽はまだ休息しているのだ。

アイハラさんちのハチが，顔を天に向けて鼻をヒクヒクさせているけれど，あいつは犬だからまだ気づいてはいまい。そう彼は思う。猫の中にもにぶいのはいる。背中に座布団をのせたようなデザインのザブなんか，おデコのハエにも気がつかないほど感度が悪い。

でも，ホンニャアにはわかるのだ。ひかえめなようすを見せながら，けっこう大きな群れをつくって咲いているアジサイの花のかげで，もう夏は光りはじめていることを……。

光る床

つゆ明けはまだ先のことなのに，気温の上昇につれて，ホンニャアの体はアメがとけるようにだんだん伸びていく。彼の体の伸び縮みは温度計のようだ。そしてわが家の木の床板としだいに仲良しになって，ダラリ，ペタリとはりついて過ごす時間がふえていく。

床張りをほどこしたものを，このごろではフローリング (flooring) としゃれた呼びかたをするらしいが，正方形をつなぎ合わせた木目の床は，夏の午後なら，猫でなくても寝そべってみたくなる。

木の性質のふしぎさは，夏はひんやりとした感触でやすらぎを与えてくれるのに，冬は冬で独特のあたたかさをただよわせることだ。どちらにしても，きれいに磨きあげておくことで，いっそう心地よさが増してくるのは，おそうじ担当者だけの満足だろうか。

毛皮をまとったホンニャアの夏はさぞたいへんだろうが，天然のクッションのような体は床にべったりとはりつくことができて，なんともうらやましい。人間ではそうはいかないし，材質のとりあわせも毛皮と床の対比にはかなわない。

湿度の高いこの午後も，ホンニャアは庭に近いリビングの床に，戸外をながめるポーズでよこたわっている。食卓の脚もとごしの，白く照り映える床に逆光のホンニャアがいて，静けさがあった。

しかし，彼のセンサーはけっして休むことはない。私がめくるかすかな紙の音や冷蔵庫のうなり声に，耳が小さく動き，しっぽが緊張する。まるで後頭部にも目があるようだ。

ふと，いたずら心がおこって，私はホンニャアにさそいをかけてみる。

「ホンニャア，コロンコロンは？」

庭を向いていたホンニャアは反射的に上半身をひねって起こし，あたりの床をキョロキョロとみまわした。

「コロンコロン」とは，ビー玉が床をころがる音の擬音なのだ。トオルが小学生のころ，床にビー玉をころがしてはホンニャアをじゃれさせて遊んだ。ホンニャア自身もビー玉との追いかけっこは好きなようだったが，私たちがあまり楽しそうなので，いっしょうけんめいサービスしているふうもあった。

「コロンコロン」の言葉は，ビー玉をころがすたびに，「ニャアちゃん，コロンコロン！」とくりかえしていたのを，いつのまにかおぼえたのだ。

もう何年も前の遊びを，ホンニャアがおぼえているだろうかと試すつもりもあったのだが，「コロンコロン」の情景は一瞬に彼のCPUからとびだしてきた。どこかの方向から光りながら走ってくるガラスの玉をさがして，ホンニャアの目もビー玉のようになった。耳には，木の床をころがるあの「コロンコロン」の音が聞こえはじめていたにちがいない。

記憶のすき間

「コロンコロン」の遊びを思いだしてしまったホンニャアは，のんびりと休むのはやめて，さがしものをはじめた。果物やワインの乗った赤いワゴンテーブルの下を，まずのぞいている。そうだ，以前はここにビー玉の入った小さな籐（とう）のカゴがあった。よくおぼえているものだ。あれをみつけたら，ビー玉遊びができると考えたのだろう。

子猫のころ，ポリエチレンの包装ひもでこしらえたボンボンが大好きで，遊びたくなると自分でくわえてきて，私たちの前にポンと投げだした。クールでわがままな彼だけれど，遊び以上に，私たちとの交流を望んでいるようすがしばしば感じられて，驚くことも多かった。

「コロンコロンをさがしてるの？」

私はホンニャアに聞いてみた。

「ウン，どこにあるの？」という目で，ホンニャアは私を見上げる。

「どこかなぁ」と，私はオーディオのラックのあたりをさがしてみせる。ホンニャアもイソイソと，私の横でいっしょにのぞきこむ。

夕飯をやるたび，「ゴハンゴハン」と語りかけていたら，とうとう猫が「ゴハンゴハン」と言うようになった話を聞いたばかりだったので，いまにそんなことが起こるかもしれない期待をかけて，ホンニャアと「会話」してみた。心がひとつになって，お互い意味することを伝えあえれば，それはき

っと会話と同じなのだ。

ところで，ビー玉はトオルの部屋にしまってあるのだから，ホンニャアには申しわけないことになった。

「コロンコロン，あるかな？」と，私はころがっているビー玉をさがすふりをして，カーテンの陰をのぞく。ホンニャアもついてきていっしょにカーテンの下に首を入れている。どうやってこの場をごまかそうかなと思っていると，ホンニャアはこんどは食器戸棚と冷蔵庫のすき間に小さな腕をつっこんでかきよせている。

細いすき間に腕のつけ根まで差し込んでいっぱいに伸ばし，つかえた顔を横向けて手の先に注意を集中しているようすがあまりにおかしい。

「あるわけないでしょ……」と思わず人間相手の調子で言いかけたとき，ホンニャアがこちら向きになって，同時にホコリまみれの丸いものがころがり出してきた。

「あらァ……」と拾いあげてみると，それはビー玉よりはあまりに小さな，オモチャのガンにつめる弾丸だった。それでも，とりあえずコロンコロンの代替品をみつけ出したホンニャアに私は敬意をはらった。

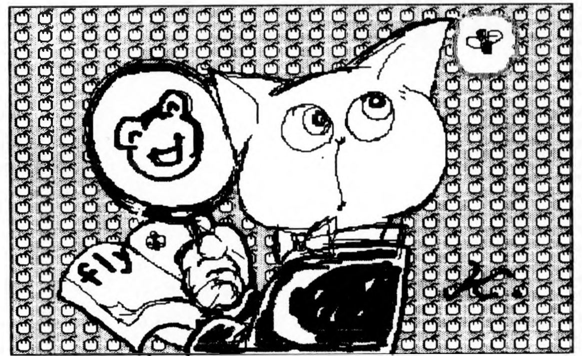
ホンニャアは自分の記憶と経験から，ビー玉のたくわえられている本拠地をたしかめてみたり，それがころがって隠れこみそうなところをいくつか推理してみた。頭の中でじっさいにビー玉をころがして，第一の候補になったのが，冷蔵庫と食器戸棚のすき間だったのだ。

さがし屋稼業

「サーチャー」という技術者が，このごろ注目を浴びはじめて，その資格を得ようとする人がどんどんふえているそうだ。

正確には「データベース検索技術者」といって，国内外のあらゆるデータベースから，必要な情報を引き出す専門家だ。基本的には，パソコン通信による各データベースへのアクセスと，必要事項の検索をするのだが，実務としての能力はなかなかたやすいものではないようだ。

「情報化社会」といままで言われてきたものも，コンピュータの活用によって，この数年でますます過密になった。現在日本で利用できる商用データベースは，海外のものが1800あまり，国内は420ほどで，5年間で4倍になったそうだ。



ある特定の「情報」を得たいと思ったとき，情報源が大きく豊富であるほど検索は複雑になり，そのための専門の知識を持った技術者が求められることになる。日本でも，そういった時代の要求から，代行検索業の会社がつぎつぎ誕生している。

そんなところで力を発揮しようという，躍進的ともいえる特殊技能のしごと，それが「サーチャー」だ。

サーチャーをめざす人のために，情報科学技術協会が昭和60年から毎年実施している，「データベース検索技術者認定試験」がある。この試験には１級と２級があって，まず２級をめざしてみんな勉強する。２級は「与えられた機器を使用して，なんとかひとりで適切な検索を行い得る能力をもつ人」(情報科学技術協会資料より)で，１級は「２級の延長上のより高度なランクであり，単に自分が適切な検索を行うことができるのみならず，初心者，２級合格者を指導，管理できる能力を持つ人」(同)だそうだ。

このサーチャーになりたい人というのが，前記の資料によると，５年前の受験者は223人，うち合格者140人，合格者のうち女性は45%。昨年度は受験者816人，合格者301人，同じく女性58%で，女性の比率が大きくなってきている。

あるサーチャー講座の教室をのぞいたら，約35人の受講者のうち女性27人，男性８人で圧倒的に女性が多く，それもおおかたは20代だった。

単なるオフィスのオペレータとはちがう特殊な技術のいる職業として，なかなか魅力はあるものの，ただのカッコよさを求めていたのでは少し甘いかなと感じる。

コロンコロンの心

認定試験の問題もなかなかのむずかしさで，パソコン通信の知識はもとより，検索のための特殊なきまり，専門用語，略語の解釈が山のようにある。その上，暗記しなければならない，あまりにたくさんのデータベースの種類，名称，特色。

空欄をうめる問題では，たとえば内容は情報検索についての一般論であっても，同義語，類似語の微妙な判別がとてもむずかしい。試験問題そのものが，検索者としての推理や分解や組み立ての能力をためしているようだ。

とはいえ，試験は正解の数が多ければいいのだ。若い人ほど暗記力はすぐれているし，「合格」はなんとかできるかもしれない。だが，そのあとの実務の世界は，マシンをあやつるだけの知識では，たやすく成り立たないらしい。

もちろんいちばんものをいうのは，各データベースの内容，特色を，自分の頭の中のデータベースにいかにたくさん取り揃えているかということかもしれない。しかし目的は，依頼者の要求にいかに適切に答えるかだ。

要求している人の目的や意図をじゅうぶんに理解する能力，その目的のために，どういう手順で検索をすすめていくかを組み立てる力。検索はかならずしもデータベースから始まるとは限らないそうだ。ときには，それ以前に「要求された情報」に関する分野の，専門家の意見が必要になることもある。そういった知己を持っていることも，検索技術者の力の一部だという。

そして，広い範囲で知識が豊かで，経験も多いこと。なによりも，インスピレーションが鋭くはたらくこと。この直感がサーチャーの腕を左右し，海外データベースへのアクセス時間も最小限にしてくれることだろう。料金も重要な条件だ。

こうしてみると，サーチャーというしごとは，人とコンピュータ，それぞれの本質を深く理解できなければつとまらない，なかなかやりがいのある新職業だ。そして，そのスピリットは，なんといっても「コロンコロン」ををさがし出したホンニャアのあのインスピレーションだ。

[第3話]

旅行あれこれ

TAKAHARA HIDEKI 高原 秀己

このところ，すっかりと海外旅行ブームは定着してしまい，もうブームなどと呼ぶのはふさわしくない。とくに年末年始や夏休みともなると，恒例行事といってもいいほどだ。日本人が海外の旅先で消費してくるお金は年間10億ドルだというからものすごい。

海外旅行にもいろいろな形式があるが，やはりパックツアーが一番の動員力を誇っているようだ。北海道や沖縄に行くのとさして変わらない金額で海外の人気旅行地に行けてしまうのだから，人気が出るはずだ。

いよいよ夏休み。

出不精のぼくも，せっかくの夏休みに何もしないのももったいないので，旅行代理店に足を運んで調べてみたのだが，パックツアーはさすがに安い。東南アジアやハワイ，グアムや東南アジアで10万円前後。15万円ちょっと出せば，アメリカ西海岸でもオーストラリアでも行けてしまう。

ところがいざ申し込んでみようとすると，なかなか難しい。

「じゃあ，この12万8000円でオーストラリアっていうの，ありますか？」

「いっぱいですね。夏休みのピークの時期のは早めになくなってしまいますよ」

それで作戦を変更して夏休みをやや外してみることにしたのだが，それもなかなかうまくいかないようだ。

「8月末出発のシンガポール・マレーシア14万8000円っていうのはどうですか？」

「まだご予約がありませんね，何人様ですか？」

「ぼくだけですよ」

「それはどうですかねぇ。おひとりですとツアーとして成立しませんので。他のお客様の申し込みを待って，ということになりますが，ご予約だけされますか？」

というわけで，旅行大作戦はひとまず延期することにして，旅行代理店から逃げ出してきた。

そもそも自分がカップルのひとりでないことが問題なのかもしれないのだが，それを気にしてはミジメになる。旅行代理店とパックツアーのシステムが，いや社会全体の歯車が狂っていることにして一件落着としてしまったのだが，この分では夏休みは今年もたいしたことはできそうにない。

どうも男性がひとりでぶらりと海外旅行をするっていうのは絵にならないようだ。そもそもがあまり，そういった不気味な客は想定されていないのだろう。

確かに雑誌でやっている旅行の特集企画にしても，ほとんど全部が女性向け。ある女性誌などは人気旅行地を毎号特集することに編集方針を変えてしまったほどだ。女性向け雑誌にはなくてはならないアイテムとなっている。

人気小説のトラベルミステリーなどにしても，たいてい事件を起こす客は女性かアベックと相場が決まっている。ひとり旅をする男性というのは刑事か探偵，あるいは出張しているビジネスマンと相場が決まっている。

かくいうぼくも，最近の旅行はスキーを除けば出張ばかり。

つい先日も，九州を数日かけて回ってきた。久々に3日以上の長さで，旅行らしい旅行だった。

仕事とはいえ地方に行くと，緑と青の自然の景色を満喫できるので，なかなかの気晴らしになる。なんせ日頃は緑といえばゴルフ場くらいしか緑のない生活を続けているのだから。

今回はキーボードから離れた生活をしたかったので，昨年末のアメリカ旅行で移動端末機として大活躍してくれたラップトップパソコン（NECの4kgのマシン）はあえて持っていかなかった。

もっともヘッドホンステレオとゲームボーイはしっかりと持っていった。この2つは退屈な飛行機や列車の中では欠かすことができない小道具だ。

九州旅行での訪問先のひとつはA社の地方工場。そこに勤務する，ある課長さんと飲みに出かけた。

その課長さん，もともとは東京本社勤務の人なのだが，ここ数年は地方工場を転々としているそうなのだ。

アルコールが十分回ってきた頃，彼はとても面白い話をしはじめた。

「妻がいうんですよ。私はA社という企業社会の中で生活しているだけなんだから，東京本社であろうが，地方工場であろうがそれほどの違いはない。ところが自分はその地域の中で生活しなきゃいけないんだから，転勤があると影響をモロに受けてしまう。だから嫌だってね」

これは盲点だった。

地方工場というのは，ロケーションこそたまたま地方にあるとはいえ，その企業の完全な一部分となって機能している。空間も工場という形で隔離されており，内部は企業社会の延長線上にある。

そこで働く人たちは県民とか町民という共通項でくくられているわけではなく，企業という名のパラレルワールドの住民なのだ。だから地方にいても，実際には地方で生活していることにはならない。

これは外資系企業のIBMとかTI（テキサス・インスツルメンツ），インテルとかを考えてみると，さらにわかりやすい。

建物のデザインや内装からしてが，しっかりとそれぞれの企業カラーが打ち出されている。内部での生活様式ならぬビジネス様式も統一されている。

入り口を通り抜ければ，もう六本木の本社の中にいるのか，地方工場にいるのかすらはっきりしないほどだ。アメリカの本社ですら，違和感はない。

これからは企業が人々の生活に占めるウエイトがますます高まってきて，国や地域の差を吸収していくという説がある。

実際にこうした地方工場の機能を見ていると，日本企業に限らず，国家とか自治体という縦割りの社会よりも強力な横割りの企業社会がジワジワと浸透してきているような気がする。

これについていける人とついていけない人とでは大きな違いが出てくるのだろう。

ちなみにその課長さん，さすがに3回目の転勤とあって，家族は東京近郊の家に戻ってしまい，哀れ単身赴任となっているそうだ。彼がいつ東京本社に戻れるのか，まったく彼にもわからないそうだ。

BACK ISSUES バックナンバー案内

ここには1989年8月号から1990年7月号までをご紹介しました。現在1989年7～12，1990年1～7月号までの在庫がございます。バックナンバーおよび定期講読のお申し込み方法については，176ページを参照してください。

1989



8月号

特集1　X1プログラミングガイドブック
PCGの基礎から奥義まで/超高速ラインルーチン 他
特集2　3Dグラフィックの深淵へ
スキャンラインZバッファ/3Dモデリング 他
新連載 (で)のショートプロぱーてぃ
X68000マシン語プログラミング/C調言語講座 PRO-68K
X-BASICプログラミング調理実習/DōGA・CGA講座
MZ-2500用グラフィックエディタ/Z80's Bar 他
全機種共通システム CP/M用ファイルコンバータ



9月号

特集　活用ハードディスク&プリンタ
各社ハードディスク接続総チェック/ハードディスク雑学講座/COPYキーメニュー/ビデオプリンタ活用プログラム 他
THE SOFTOUCH ジェノサイド/琉球/mFORTH Compiler
●サイバースティックで遊ぶ **不思議な環境ソフトの世界**
●X1/X1turbo用シューティングゲーム **Defeat X**
Z80's Bar/MZ-2500グラフィックエディタ 他
[X68000] X-BASIC/マシン語/C調言語講座/DōGA・CGA
全機種共通システム 生物進化シミュレーションBUGS



10月号

特集　ゲーム面白心理学
ソーサリアン・宇宙からの訪問者/ファンタジーゾーン
ねじ式/ガウディ・バルセロナの風/サバッシュ 他
●MZ-700用シューティングゲームSide Roll-F
●X1/X1turdo用カードゲームBonding
ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/DōGA・CGA
THE SOFTOUCH Z'sTRIPHONY DIGITAL CRAFT/James68K
全機種共通システム 小型インタプリタ言語TTI



11月号

特集　microComputer入門
初歩からのCPU物語/RISCプロセッサの設計と製作
X68000&X1で周辺LSIを使いこなそう
連載 ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/DōGA・CGA
●X68000用カードゲームばばぬき
LIVE in '89 メタルホーク/オブ・ラ・ディ，オブ・ラ・ダ
THE SOFTOUCH Stationery PRO-68K/リングマスター1
全機種共通システム TTI用パズルゲームPUSH BON!



12月号

特集　Cプログラミングへの招待
付録　C言語簡易リファレンス
連載 ショートプロぱーてぃ/Z80's Bar
X68000マシン語/X-BASIC/DōGA・CGA
●Oh!X2周年特別企画「素粒子の声が聞こえる」
●X1/turbo用アクションゲームACTIVE UNIT
LIVE in '89 天空の城ラピュタ/ギャラクシーフォース
THE SOFTOUCH 38万キロの虚空/た～みのる2
全機種共通システム SLANG用リダイレクションライブラリ

1990



1月号

特集1　オペレーティングスタイルの研究
特集2　Cプログラミング応用編
連載 ショートプロぱーてぃ/Z80's Bar
X68000マシン語/C調言語講座/DōGA・CGA
●X1/turbo用シミュレーションゲームSuper Battle
LIVE in '90 さよならを過ぎて/RYDEEN
THE SOFTOUCH レナム/メタルサイト
全機種共通システム WORM KUN/再掲載SLANG
特別付録 X68000 THE SOFTWARE CATALOGUE



2月号

特集　画像圧縮へのアプローチ
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X68000マシン語/C調言語講座/X-BASIC調理実習
●X68000用ゲームプログラムGon Gon
●MZ-700用紙芝居Eyelarth
LIVE in '90 オーダイン/魔女の宅急便
THE SOFTOUCH A-JAX/フラッピー2/夢幻戦士ヴァリスII
マジックパレット/Mu-1/CYBERNOTE PRO-68K
全機種共通システム 超小型コンパイラTTC++



3月号

特集　MUSICアドベンチャー
X68000用MIDIドライバ&音源エディタ
なんでも鳴らせるOPMD.X/MMLを楽譜データに
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
C調言語講座/X-BASIC調理実習
●X1/turboシミュレーションCRISIS in Tokyo
LIVE in '90 パワードリフト/スキーム/となりのトトロ
THE SOFTOUCH ナイトアームズ/斬/ダンジョンマスター
全機種共通システム 超多機能アセンブラOHM-Z80



4月号

特集　ゲームシステム文学誌
1989年度GAME OF THE YEAR発表
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/C調言語講座/X68000マシン語
●X1・MZ-2000/2500用RPG The Cave of Dalk
●うわさの68040，ついに登場
LIVE in '90 バーニングフォース(OPMD対応)
THE SOFTOUCH The Fille Professor/HOST PRO-68K
全機種共通システム ファジィコンピュータシミュレータI-MY



5月号

特集　BASICプログラミング
第5回　言わせてくれなくちゃだワ
連載 ショートプロぱーてぃ/Z80's Bar
X-BASIC調理実習/X68000マシン語プログラミング
●新機種X68000SUPER-HD/EXPERTII/PROII
●ラジコンスティックの製作
LIVE in '90 TURBO OUTRUN
THE SOFTOUCH 天下統一/ポピュラス/Hyperword
全機種共通システム インタプリタ言語STACK



6月号

特集　創刊8周年記念PRO-68K(付録5"2HD)
Oh!Xアンケート結果大分析大会
連載 ショートプロぱーてぃ/Z80's Bar/PurePASCAL
X-BASIC調理実習/X68000マシン語プログラミング
●X1turbo用コマンドシェルシミュレータ
●ハードウェア工作入門
LIVE in '90 ナイトアームズ/悪魔城伝説/この木なんの木
THE SOFTOUCH 三国志II/FAR SIDE MOON/グラナダ
全機種共通システム X68000用S-OS"SWORD"他



7月号

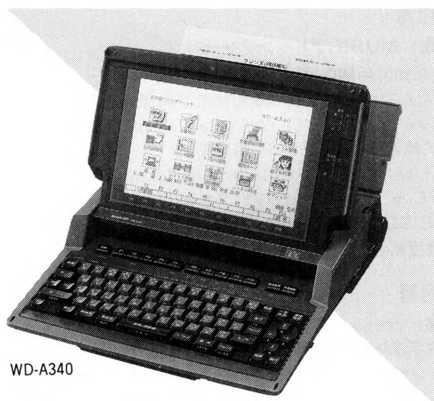
特集　マシン語への第一歩
X68000SUPER-HD試用レポート
連載 ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/PurePASCAL
●INTEGRAL X1――ノーマルX1への対応
●ハードウェア工作入門
LIVE in '90 夢幻戦士ヴァリスII/トッカータとフーガニ短調
THE SOFTOUCH サーク/あーくしゅ/ダウンタウン熱血物語
全機種共通システム リロケータブルアセンブラWZD

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

スーパーアウトラインフォント内蔵
WD-A320/340
シャープ



WD-A340

シャープは「見やすい大型液晶画面」,「活字に迫る高品位印刷」,「思いどおりのレイアウト」,「正しいことばづかい」などを追求したラップトップ型ワープロ「WD-A320」および「WD-A340」を発売した。

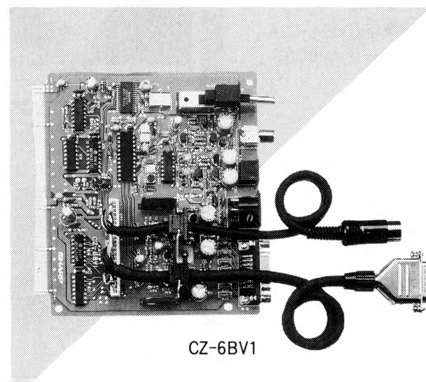
「WD-A320/340」は新開発の専用LSIにより名刺用の小さな文字から拡大文字まで美しくなめらかに印字する,「書院スーパーアウトラインフォント」を内蔵している。曲線データで文字を形成しているため,直線(ベクトル)データによるアウトラインフォントに比べ品位を向上している。4.5~288ポイントまで合計67種類のマルチポイント文字(欧文時はマルチポイント23種類)を自由に設定することで,多彩な大きさの文字を利用できる。また,それに加えて64ドット・400DPIの高精細プリンタを搭載していることで,美しい印字が可能となっている。

さらに,パーソナルDTP機能,手紙文の作成に便利な「直子の代筆(書院版)」,15万例のAI-V3辞書,電子手帳とのデータの共有ができる電子手帳機能などの機能も装備している。

「WD-A340」ではこれに加えてハイコントラスト白黒液晶画面,類語辞書,文体統一機能などの文書校正支援機能,MS-DOSコンバータ,通信ソフトなどを搭載している。価格はそれぞれ178,000円と198,000円(どちらも税別)。

〈問い合わせ先〉
シャープ㈱ ☎06(621)1221,03(260)1161

X68000用ビデオボード
CZ-6BV1
シャープ



CZ-6BV1

シャープはX68000用の周辺機器としてビデオボード「CZ-6BV1」を発売した。このボードをX68000の拡張I/Oスロット(2スロット分を使用)に装着することにより,コンピュータ映像をビデオ信号として取り出すことができるようになる。たとえば,X68000上で作ったグラフィックやアニメーションあるいはゲーム画面などを手軽にVTRに録画することができる。さらに,ビデオ入力端子のついている液晶ビジョンや大型テレビにX68000を接続して,迫力ある大画面でゲームなどを楽しむこともできるようになる。特徴は以下のとおり。

・NTSCエンコーダ,同期信号発生回路とも1チップ化
・入出力端子は以下のものを装備
　アナログRGB×2
　テレビコントロール×2
　S映像出力×1
　コンポジットビデオ出力×1
・高解像度モード時のビデオ出力を自動的に停止することができる

価格は21,000円(税別)。

〈問い合わせ先〉
シャープ㈱ ☎06(621)1221,03(260)1161

X68000とMacをリンク
Mac版「XIN/XOUT」
電机本舗

電机本舗はRS-232Cを介してデータ転送をするシステム,「XIN/XOUT」のMacintosh版を発売した。これはRS-232C/422通信ポートを利用して,Macintosh Plus, SE, SE/30, IIとMS-DOSマシン/X68000の間でのファイル転送を可能にするものである。バイナリファイルの転送も可能で(エラーチェックは独自のものを採用),ファイルの一括指定一括転送もサポートしている。転送に際しては,転送先のファイル形式に自動変換,OSの相違を完全吸収し漢字を含んだファイルも正確に転送する。英語,日本語環境およびマルチファインダ上にて動作する。

パッケージにはRS-232Cケーブルと,ファイル転送プログラムのMac版とMS-DOS(/X68000/PC-DOS)版のフロッピーディスク2枚が入っている。価格は12,800円(税別)。

〈問い合わせ先〉
㈲電机本舗 ☎03(447)1773, BBS 03(447)2564 1200bps

XIN/XOUT



電子手帳用プリンタ&名刺管理カード
CE-80P,PA-7C50/7C51
シャープ



シャープは既存の電子手帳すべてに接続可能なプリンタ「CE-80P」を発売した。さらに，面倒な名刺の整理に便利な名刺管理カード「PA-7C50/51」を7月25日に発売する。

電子手帳用プリンタ「CE-80P」ははがきやラベルへの宛名印字はもちろん，リフィルへの住所録印字もできる。別売のはがきフィーダを装置すれば，連続20枚までのはがき裏面の連続印字が可能。年賀状などで使うあいさつの慣用句73種類を内蔵しており，また，オプションの毛筆体カートリッジ「CE-61M」により美しい毛筆体での印字が可能になるので年賀状などが簡単に作成できる。リボンカセットは黒，赤，青，茶，金，銀が用意されていて（茶は8月発売予定），6色印字が可能。価格は45,000円(税別)。

名刺管理カード「PA-7C50/51」は名刺情報はもちろん，いつ，どんな用件で会ったのかを記憶できる交際録，趣味や嗜好を記憶できる備考，年賀状やお歳暮などの状況をチェックできるチェックリストなどの記憶が可能。名刺情報は名前4文字，電話番号12桁，FAX番号12桁，会社名8文字，所属5文字，役職2文字，郵便番号3桁，住所20文字の場合で約350人分（PA-7C50の場合は約160人分）が記憶できる。機能としては郵便番号辞書，日付検索やチェック検索などの多彩な検索機能，宛名印字機能を搭載。さらに本体メモリをバックアップできるRAMファイルとしての使用も可能となっている。価格は「PA-7C50」が13,000円，「PA-7C51」が16,000円。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221,03(260)1161

32ビット浮動小数点DSP
DSP96002
モトローラ

モトローラは24ビット固定小数点デジタルシグナルプロセッサDSP56000ファミリの上位機種として，32ビット浮動小数点DSP96002を開発した。

・動作周波数：27MHz,33MHz
・命令サイクル：74nsec,60nsec
・IEEE754データフォーマットに準拠
・43×43ビット→96ビット浮動小数点演算
・32×32ビット→64ビット整数演算
・12Gワードのメモリ空間
・1Kワードのオンチップデータ RAM
・1Kワードのオンチップデータ ROM（サイン，コサインテーブル）
・512ワードのオンチッププログラム RAM
・2チャンネル DMAC
・32ビットバレルシフタ
・223ピンセラミック PGA パッケージ
・割り算と平方根用に高速な命令（6命令サイクルと9命令サイクル）を用意

DSP96002の2つの外部メモリ拡張ポート（ポートAおよびポートB）はユーザープログラミングによって，外部メモリのアクセスポートあるいはホストプロセッサとの接続ポートとして使用できる。さらに，DSP96002の各ポートにはマルチプロセッサ構成をサポートする信号線も用意されているので，複数個のDSP96002でマルチプロセッサを構成し高性能な演算処理を実現することもできる。

以上のような特長により，DSP96002は従来のDSPでは処理が困難であった画像処理，浮動小数点演算アクセラレータ，医用機器，周波数解析処理などに応用が可能である。

〈問い合わせ先〉
モトローラ㈱　☎0120-068030

INFORMATION　番外編

「X68000グッズショップ in Akihabara」
ミナミ電気株式会社　本館5階

X68000グッズが買いたいと思っても，いままでは常備店がなかったので，イベントに行って買うなどしか方法がありませんでした。しかし，このたびミナミ電気本館5階のパソコンフロアにX68000グッズショップ in Akihabaraが開設されることになり，いつでもX68000グッズを手に入れることができるようになりました。

そこで，それを記念してひょっとしたらあまり知られていないかもしれないグッズの数々を紹介してみたいと思います。

★X68000牛革ベルト
標準価格6,300円（税別）
バックルには光輝く“X”のロゴが……
★X68000キーホルダー
標準価格1,300円（税別）
X68000の電源スイッチにも鍵があればよかったのに
★X68000ネクタイピン
標準価格3,000円（税別）
ネクタイをする人にはいいかも
★X68000電飾POP
標準価格9,500円（税別）
暗い所で見ると本当にきれい
★X68000クリスタルポルシェ
標準価格8,000円（税別）
ガラスでできたポルシェ911
★X68000ジッポ・ライター
標準価格4,800円（税別）
あのツタンカーメンの仮面が……
さらに，
★X68000ゴルフボール
標準価格1,900円（税別）
★X68000傘
標準価格4,200円（税別）
★X68000スポーツタオル
標準価格3,300円（税別）
と，「こんなものまで？」と思うような変わった(?)商品が，ほかにもまだまだいろいろあります。興味のある方はお店でご覧になるとよいでしょう。
☆万世橋交差点際　第一家電隣

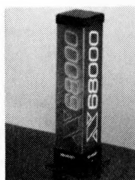
牛革ベルト



キーホルダー/タイピン



電飾POP



ジッポ・ライター



ゴルフボール



FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。毎日暑い日が続きますね。夏バテや寝冷えに気をつけて，楽しく有意義な夏休みを過ごしてください。

参考文献
I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
The BASIC　技術評論社
テクノポリス　徳間書店
POPCOM　小学館
マイコン　電波新聞社
マイコンBASIC Magazine　電波新聞社
LOGIN　アスキー

一般

▶特集シムアース
シムシティーの登場によって示されたパソコンシミュレーションの楽しさ。今度はもっとグローバルに地球環境のシミュレーションをやってしまおう。そこで発表されたのが「シムアース」。その概念や裏話などを解説。シムアースを考える座談会にはミュージシャンの細野晴臣，戸田誠司，日本自然保護協会の横山隆一らが参加している。——編集部，LOGIN，12号，116-127pp.

▶ネットワーカー・ホリック　第22回
新聞の申し込みまでできちゃうぞ。大手ネットのショッピングサービスを紹介。PDSはPC-9801のZMODEM転送プログラム「ZM. EXE」，X68000のシューティングゲーム「MEMORY BROKEN.X」。全国BBS探訪記は秋葉原にあるPENCIL-NET。——編集部，LOGIN，12号，202-203pp.

▶ハードラボラトリー
MIDIについて解説。X68000の純正MIDIボードCZ-6BMIやMusicstudio PRO-68Kも紹介。——編集部，POPCOM，7月号，106-108pp.

▶X68000のウイルス騒動の真相
先頃新聞を騒がせたX68000用市販ソフトへのウイルス混入事件についてウイルス騒動の当事者が内情を語る。日コン連では昨年11月に各マスコミへ今回のウイルスのソースリストを送っていたという。——日コン連理事長山本隆雄，The BASIC，7月号，176-177pp.

▶２大ショウに見る最新パソコンの現状
ビジネスショウ・マイコンショウに展示された各社の新製品をレポートし，今年のトレンドを探る。——編集部，マイコン，7月号，135-144pp.

▶コンピュータ・ウイルスを考える
ウイルスについて正しい理解をするために，ウイルスの種類や事例，対策について述べる。——コンピュータ・ウイルス研究会，マイコン，7月号，164-165pp.

▶楽器が弾けなくても，声で楽器が演奏できる
マイクロコンピュータショウに出展されていた，ボイスインプッタを紹介。マイクに入力された音程を解析してMIDI楽器を鳴らすことができる。——FORESIGHT企画部・藤本健，マイコン，7月号，239-240pp.

▶ビジネスマンの情報管理術
著者のヨーロッパ旅行記第３弾。ポルトガル，オランダ，イギリスなどで7カ国語翻訳カードと通貨換算機能が活躍する。——塚田洋一，マイコン，7月号，310-312pp.

▶やまさんのアルゴリズム・ブック
MS-DOSなどで頻繁に使われるワイルドカード機能のアルゴリズムを考える。——やまさん，マイコン，7月号，321-325pp.

▶実践ハード入門
梅雨にあわせて，湿度センサを使った簡易湿度計を作る。——石川至知，マイコン，7月号，334-336pp.

▶レーザーディスクで拡がるマルチメディアの世界
レーザーディスクの生み出すハイパーメディアの世界について述べ，またマッキントッシュでのハイパーメディアの現状を報告する。——田島恵介・長谷川昌夫，マイコン，7月号，346-354pp.

▶NEW MACHINES '90
NEC，エプソンなどの新機種と共に，AX仕様のAll in Note，X68000SUPER-HDを取り上げ，概要を紹介する。——編集部，ASCII，7月号，258-280pp.

▶AtariSTの魅惑の世界
68000使用のホビーパソコン，米Atari社のSTシリーズの魅力に迫る。今月はラインナップ，ハードウェア，PDSやゲーム事情などについて。——小沢靖・池田賢司・判治聡，ASCII，7月号，313-320pp.

▶MEDIA BREAK
北九州市八幡にオープンしたスペースワールドの宇宙飛行士訓練プログラム「スペースキャンプ」を紹介。——浦山明俊・佐藤守弘，ASCII，7月号，409-411pp.

MZシリーズ

MZ-1500（MZ-5Z001 BASIC）

▶1582
カプコンのシューティングじゃないよ。戦国アクションゲーム。——大石豊，マイコンBASIC Magazine，7月号，126-128pp.

MZ-2500（BASIC-M25）

▶BLOCK BROKEN
ブロックと入れ替わる難解パズルゲーム。——TaK KuN，マイコンBASIC Magazine，7月号，129-130pp.

▶Multi Window
BASICのウィンドウサブルーチン。——佐藤拓也，マイコンBASIC Magazine，7月号，179-180pp.

X1/turbo/Z

X1シリーズ

▶最新ゲーム徹底解剖!!
新着ゲーム「スライミャー」の基礎攻略法を紹介。——編集部，LOGIN，11号，226-227pp.

▶攻略おすすめゲーム
ウィザードリィVの地下３階までを攻略。——編集部，テクノポリス，7月号，50-53pp.

▶桃四郎
好評の桃シリーズ，今回は桃太郎４人目の兄弟の話。お供をやとい鬼をたおすアクションゲーム。ジョイステ

新刊書案内



この人の著書（「ハイパーメディア・ギャラクシー」など）を読むと，実に「2001年宇宙の旅」に関する話が多い。趣味が高じてか今度は「2001年宇宙の旅」を中心においた映像論の本を書いた。本書は2つの点で実に面白い。ひとつは，そこいらの映画評論家が書く映画評より資料も視点もしっかりしていること。もうひとつは，どうして著者はコンピュータはメディアを目指すべきだと考えるのか。メディアとなったコンピュータに何を期待するのかがはっきりとわかることだ。

オーソン・ウェルズ，小津安二郎，そしてキューブリックの3人の映画監督の共通点。彼らは何と戦い，何を表現しようとしたのかということ。「ジョージ・ルーカスやスティーブン・スピルバーグは，最新の特撮技術を総動員して，過去のイメージを増幅しているだけ」だということ。HALはなぜ殺さねばならなかったのかということ（2010年で示されたような安易な答えではない）。「2001年宇宙の旅」はメディア論だということ。著者はメディアとしてコンピュータを使うことによって，個人の表現を復権させたいのである。　(K)

キューブリック・ミステリー　浜野保樹著　福武書店
☎03(230)2131　新書判　204ページ　1,130円

ィック専用。――ズオ，マイコンBASIC Magazine，7月号，158-160pp.
▶LEADER LEADER
シルクハットをかぶったハット君にパンを食べさせてゴールに向かう。風船で道をつくってハット君を誘導する。風船パズルゲーム。――吉川章，マイコンBASIC Magazine，7月号，161-164pp.
▶性格判断
学園祭の定番，性格判断プログラム。多少判定の文章が貧しいという声もなくはないが……。――編集部，マイコン，7月号，212-216pp.

X1+FM音源ボード（要NEW FM音源ドライバ）
▶ミスティ・ブルー
エニックスのアドベンチャーゲームのミュージックプログラム。――KENJI，マイコンBASIC Magazine，7月号，192-194pp.

X1 turboシリーズ
▶NEW SOFT
セレクテッドソーサリアン4のシナリオの解説。――編集部，LOGIN，12号，12-13pp.
▶攻略おすすめゲーム
世界の海を股にかけるゲーム，「大航海時代」の最も重要な要素，交易について攻略。――編集部，テクノポリス，7月号，46-49pp.
▶月に帰りたいヒトデちゃん
降ってくる星を足場にして月まで帰る。スクロールアクションゲーム。――HARU，マイコンBASIC Magazine，7月号，164-165pp.

X68000

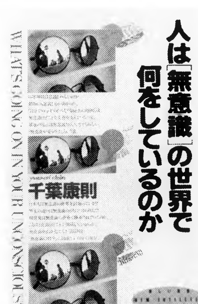
▶NEW SOFT
7月発売予定の「ウルティマV」と「闇の血族」，そのほか発売中の「パズニック」「天下統一」「ダウンタウン熱血物語」を紹介。――編集部，LOGIN，11号，12-25pp.
▶X68000新聞
戦国ゲーム特集。「天下統一」をはじめ「信長の野望・全国版/戦国群雄伝」「斬〔ZAN〕」を紹介。そのほか「POOL BAR」「闇の血族」「ダウンタウン熱血物語」「ガンシップ」を紹介。――編集部，LOGIN，11号，162-167pp.
▶最新ゲーム徹底解剖!!
新着アクションゲーム「グラナダ」の攻略・その2。ステージ4からステージ6までを，マップを載せて紹介。アクションパズルゲーム「スライミャー」も紹介。――編集部，LOGIN，11号，196-199・226-227pp.
▶Software Review
ポピュラスを真面目に考えてみる！　ほかのゲームとはちょっと違うポピュラスの面白さとは？　――川村B，LOGIN，11号，230-231pp.
▶NEW SOFT
8月発売予定のシミュレーションゲーム「JOSHUA」，7月発売予定の「POOL BAR」を紹介。――編集部，LOGIN，12号，19・22p.
▶X68000新聞
新着ゲームの紹介。「ラグーン」「維新の嵐」「ルーンワース」。そのほかジェノサイドのCDレコーディング風景やThe File Professorの解説。――編集部，LOGIN，12号，130-135pp.
▶先取りおすすめゲーム
7月中旬発売予定の「ラグーン」を紹介。――編集部，テクノポリス，7月号，14-15pp.
▶GAMING WORLD
好評のくにおくんシリーズ「ダウンタウン熱血物語」，アクションパズルゲーム「パズニック」「スライミャー」「タッグ・オブ・ウォー」，発売予定の「ユニオン」「レインフォーサー」「RYU～哭きの竜より～」を紹介。――編集部，テクノポリス，7月号，18-30pp.
▶攻略おすすめゲーム
第二次大戦のフランス戦をあつかった陸戦シミュレーションゲーム「機甲師団」を徹底攻略。――編集部，テクノポリス，7月号，56-57pp.
▶レモンちっくWORLD
発売予定の美少女RPG「ランス2～反逆の少女たち～」，麻雀ゲーム「ぴんぴん麻雀ピーチエンゼル」，カードゲーム「DOKI DOKI Card League」を紹介。――編集部，テクノポリス，7月号，72-79pp.
▶SLGの夏が来た!!
シミュレーションゲーム特集。ポピュラスの紹介やその原作者ピーター氏からのありがたいお告げなど。――編集部，POPCOM，7月号，62-63pp.
▶WE ARE THE X68000 WORLD IN HOKKAIDO
新着ゲーム「ラグーン」「POOL BAR」「Vessel」「サーク」「ルーンワース」「レインフォーサー」「ユニオン」などとスプライトツール「ぴくせる君」を紹介。――編集部，POPCOM，7月号，68-72pp.
▶ゲームがオレを呼んでいる！
くにおくんシリーズ「ダウンタウン熱血物語」と発売予定のゲーム「ウルティマV」の攻略法を解説。――編集部，POPCOM，7月号，82-90pp.
▶パズルDEバトル
新着パズルゲーム「パズニック」を紹介している。――さすらいのパズラー，POPCOM，7月号，92-93pp.
▶ミュージックパビリオン
映画「香港パラダイス」の主題歌「無敵のビーナス」（GO-BANG'S）のミュージックプログラム。――編集部，POPCOM，7月号，176-179pp.
▶キミのX68000を護れ！
コンピュータウイルスの基礎知識ほか，X68000のIPL，SRAM常駐型ウイルスに対して有効なワクチンソフトを誌上公開。――GORRY，マイコンBASIC Magazine，7月号，67-73pp.
▶誌上公開質問状
X-BASICの画像フォーマット「GL3」の解説や，カラーイメージユニット「CZ-6VT1」の機能紹介。そのほかCommunication PRO-68KでATモデムは使えるか？　などの質問に答えている。――多田太郎，マイコンBASIC Magazine，7月号，90p.
▶わかった！
画面に隠れたアルファベットを当てる。マウス専用，文字さがしゲーム。――小野正明，マイコンBASIC Magazine，7月号，166-167pp.
▶PYRAMID BREAK
ピラミッド型につまれた5種類のブロックを落とさずに取っていく。山くずしゲーム。――萬道賢治，マイコンBASIC Magazine，7月号，168-170pp.
▶リレーレビュー
ウルフ・チームの「グラナダ」について，4人のライターの意見を聞く。――編集部，マイコン，7月号，194-195pp.
▶スクリーンエディタEDX
Human68kとOS-9/X68000上で共通の操作環境を提供するスクリーンエディタ。いわばED.Xの機能強化版である。――村田誠，ASCII，7月号，335-338pp.
▶AV STRASSE
PDSのグラフィックエディタ，MFGEDを紹介。高機能ではないが瞬時に立ち上がる小回りの良さが身上。――仲田津弘，ASCII，7月号，353-356pp.
▶NEWBAT.X
以前発表されたBATKEY.Xのバージョンアップ版。バッチファイルの機能を拡張してくれる。――牛島健雄，I/O，7月号，198-202pp.
▶迷路エディタ
最大511×511のマス目にマウスで絵を描くと，それを正解として迷路を作ってくれるというもの。――カバウシ2世，I/O，7月号，189-197pp.

ポケコン

PC-E500
▶TURBO RUN
ドライビングゲーム。――森高周作，マイコンBASIC Magazine，7月号，175p.
▶DRAGON BUSTERD
ドラゴンバスターことクローブスを操作してドラゴンをやっつける。アクションゲーム。――広鹿太一，マイコンBASIC Magazine，7月号，176-177pp.



エッシャーからの贈り物
エッシャーの描いた数々の作品を，CGで表現した。同じ内容のビデオも発売されており，そちらのほうがメインのようだ。作品の質としては今ひとつの感があるが，ビデオで見るとまた違った味わいだろう。エッシャーの騙し絵をCGにしちゃおうという発想はなかなかよい。　(K)
野崎昭弘著　小学館
☎03(230)5442　B5判　47ページ　1,680円



人は「無意識」の世界で何をしているか
無意識の世界。カッコよくいうと，サブリミナルとか潜在意識とかとなる。本能や反射など，とにかく，人間のほとんどの活動は意識に現れないところで行われている。自分は意志に基づいてのみ行動していると思っている人，これを読んで謙虚になりなさい。PHPくさいところがわずかにあるが，丁寧な語り口で脳と無意識と行動の話を紹介している。専門的な内容はほとんどない。わからないことはわからないとしているのも善良。(K)
千葉康則著　PHP研究所
☎03(239)6221　B6判203ページ　1,000円

QUESTION and ANSWER



X68000のアセンブラで乱数発生のプログラムを組もうと思うのですが，乱数発生の原理がわからず困っています。乱数発生の原理（乱数は1ロングワードの整数）はどうなっているのでしょうか？　徳島県　森上　晶仁

一般に乱数は線形合同法と呼ばれる方法で作られています。これはある式に値を代入して計算によって乱数を生成する方法で，詳しい説明が1988年8月号に紹介されていますから興味のある方はそちらをどうぞ。

ところで，X68000には乱数を生成するためのファンクションコールが用意されていますから，それを利用することにして使い方を説明しましょう。

まず，このファンクションコールはFLOATn.Xを組み込むことによって使えるようになるものです。乱数発生部のコール番号は＄FE0Eとなっていますのでアセンブラで書くなら，

```
dc.w $FE0E
```

もしくは，FEFUNC.Hをインクルードして，

```
FPACK RAND
 (戻り値はd0.w)
```

という具合に使うことになります。

また，乱数系列の初期化には，

```
dc.w $FE0D
FPACK SRAND
 (引数はd0.w)
```

とします。内容はBASICのRAND( )，SRAND( )と変わらないと思います（たぶん）。

ここで得ることのできる乱数の値の範囲は，0から32767と森上さんの希望とは違うものですが，実際には32ビットの乱数を必要とされることは稀だと思いますし，もし必要なときはこの方法で得た乱数にビットシフトなどの加工をしてから，さらに乱数を加えるとか，工夫次第でどうにでもなるでしょう。



編集室の皆様こんにちは。僕は2年たってもろくにプログラムの組めない大バカ野郎です。6月号の付録のディスクはとてもよかったです。大事に使わせてもらっています。僕は前からCGをやってみたいと思っていました。だからANGELが動くのを楽しみにしていたのです。

いざ解凍してみてコマンドモードで“ANGEL”と入力してみると，「主記憶が足りません」と出てきました。ASK68Kをはずしてみなさいと書いてあったので，自分なりにはずしてみましたが同じメッセージしかでてきません。もう一度ASK68Kをはずすところからできるだけ詳しく書いてください。機種はX68000ACE，Human68k Ver.1.01，メインメモリは1Mバイトです。

愛知県　藤田　聡

同じ内容の質問がほかにも何通か送られてきましたが，藤田さんのハガキが一番最初に送られてきました（往復ハガキは使わないでくださいね）。とにかくX68000というマシンはメモリを大量に必要とするマシンです。標準で1Mバイトしか積んでいないマシンを使っている方は，BASICから子プロセスを実行することもままならないでしょう。

普通に考えれば，メモリを増やすにはパソコンショップにいって増設メモリを買ってこなくてはいけませんが，とりあえず使うことのないデバイスドライバを組み込まないようにしてメモリの空き容量を増やすことも可能です。質問電話によると藤田さんと同様のケースではほとんどがビジュアルシェルから起動したためのメモリ不足でした。このあたりの話は先月号でも触れられていましたが，もう少し詳しく話しましょう。

Human68kは起動したドライブに存在するCONFIG.SYSの内容に従ってデバイスドライバの組み込みを行います。つまりASK68Kなどのデバイスドライバを組み込まないということは，CONFIG.SYSの内容を変更することにほかなりません。それにはエディタ，ワープロ，またはCUSTOM.Xのどれかを使うことになりますが，ここではエディタを使って変更するとしましょう。まず，

```
ED A：¥CONFIG.SYS
```

としてCOFNIG.SYSをエディタに読み込みます。この場合はED.Xがパスの通っているディレクトリにあり，CONFIG.SYSがドライブAのルートディレクトリ上にあるものと考えています。画面のどこかに，

```
DEVICE＝¥SYS¥ASK68K.SYS…
```

といった行があるはずですから，それを

```
*DEVICE＝¥SYS¥ASK68K.SYS…
```

と先頭に＊を挿入します（＊をつけると注釈行扱いとなる）。こうしてからESC･Eでファイルをセーブしてエディタを終了させます。これでASK68Kを組み込まないシステムの完成です（注：リセットして再起動しなくてはいけません）。

ほかにも登録したくないデバイスドライバがあったら，同様の変更をすることで組み込まないようにすることができます。プリンタドライバやPCMドライバもとりあえずいらないでしょうし，間違ってもRAMディスクを設定してはいけません。

また，Human68k Ver.2.0などには，

```
OPMDRV.X
HISTORY.X
FLOATn.X
IOCS.X
```

など，実行可能ファイルのくせにデバイスドライバとして登録できるものがあります（このうち，必ず設定しなければならないのはFLOATn.Xのみです）。FM音源を使うならOPMDRV.Xをデバイスドライバとして登録するために，

```
DEVICE＝OPMDRV.X
```

と書くことになっていますが，そうしなくともコマンドモードから，

```
A：¥SYS¥OPMDRV
```

とすれば，FM音源を使うことができるし，

```
A：¥SYS¥OPMDRV OFF
```

とすれば，いつでもFM音源を使わないようにすることができます（使えなくなるだけで空きメモリが増えるわけではない）。

ですからFM音源を使うことが滅多にないのなら，OPMDRV.Xを組み込まないようにしたほうがいいでしょう（標準1Mバイトの方は特に）。OPMDRV.Xを使用する

ソフトを起動したときは，エラー($FE0D)が発生しますから，そしたらOPMDRVとコマンドモードから入力すればいいのです。こうしておけば，OPMDRV.Xを使わない場合は通常87000バイト，コマンドモードから登録した場合も，わずかですが3000バイトほど空き容量が多くなります。また実行速度も割り込みが発生しない分だけ，いくらか上がります。

またIOCS.Xを組み込んでいる人もメモリが狭いと感じるようだったらはずしておくことをすすめておきます。スクロールの高速化などあれば便利ですが，なくても動くんだから我慢しましょう。また，FILESやBUFFERSの最初の数字も小さくすると多少はメモリ消費が抑えられます。ディスクアクセスが遅くなったり，同時に扱うファイル数に制限が出ますが「背に腹は代えられぬ」ってやつですね。

もちろん，このような操作も，ビジュアルシェルで起動すると台なしです。真っ先にコマンドシェルで起動するシステムディスクを作ってください。方法は各機種の取扱説明書第3部「より高度な使い方」の3章「デスクトップを使わない操作」の4項「起動時にコマンドモードに入るには」を参照してください。



パソコンの画面をビデオに録ろうと思い，X68000のカラーイメージユニットを買ったのですが，市販のソフトウェアをビデオに録るときに，コンピュータの画面モードをスーパーインポーズすると黒が透けてテレビ番組が映ってしまいます。VCUTを実行しようとしても市販ソフトなので無理ですのでどうしようもありません。どうにかテレビ画面をカットする方法はないでしょうか。

静岡県　石井　孝



スーパーインポーズの状態でないとビデオ録画できないという制約がなければなんでもないことなのですが，どんなに考えてもスーパーインポーズさせないと録画できないのは仕様上，変更することは無理だと判断できます。

問題点はスーパーインポーズにあるのではなく，黒色が透明扱いされてテレビ番組が映ってしまうことなんです。

ということは，もしチャンネルをあわせたときに画面全体が真っ黒な放送があるとして，そこでスーパーインポーズしたらどうなるか。……そうですね，コンピュータ画面の黒（透明色）の部分にビデオ信号の黒が入って，うまくコンピュータ画面がそのまま録画できるわけです。

ところが，そんな放送があるわけがないので，どうやって黒色の画像を手に入れるかが問題となってきます。しかも，それを通してコンピュータ画面を見るのですから，ノイズの多いビデオ信号だと録画したときに画像が乱れて見にくいかもしれないので，できるだけ安定したものを探すことになります。

私の知っているものではセガマークIIIやメガドライブ，PCエンジンなどのゲーム機のカセットを入れずに電源を入れると，真っ黒の画面が流れたように記憶しています。ただし，これらは正確にはビデオで使っているビデオ信号とは微妙に異なる場合があるので，もしかしたら同期がずれたりノイズが出る可能性もあります。結局は手持ちのビデオ機器との相性次第ですので注意してください（録画側のビデオデッキにTBC機能がある場合はTBCをON/OFFして相性を調べてください）。

また，2台以上のビデオデッキがある場合，ほとんどのビデオデッキが外部入力にして画像を入力しなければ，画像出力側には真っ黒（灰色？）な映像信号が流れると思います。それらの出力をカラーイメージユニットのビデオ入力につなげておいてスーパーインポーズすれば，うまく録画できるでしょう。

なお，近日発売が予定されているビデオボード（カラーイメージユニットの録画専用版，イメージ取り込み機能はない）では内部にビデオ信号発生機を持っているのでこのような面倒な操作は必要なくなったようです。すでにカラーイメージユニットをお持ちなら特に必要ないと思いますが。



Oh!X1988年9月号のturbo RAY TRACERが動きません。リストを同封しますので，おかしいところがあれば教えてください。

北海道　村松　良彦



村松さんの質問は便箋2枚にわたる長いものだったので，質問を簡略化させてもらいました。ところで，送られてきたリストと質問の内容から判断すると，こちらの説明不足のため動作していない可能性もありますので，一応補足説明させてもらいます。

記事ではリスト6からリスト9がデータの例として掲載されていますよね。これらのデータはリスト3のデータセットプログラムにマージして使うようになっているのですが，そのことが記事の中で触れられていません。たとえば，リスト6の例1が「EXAMPLE1」として保存してあるのなら，リスト3をロードしたあとに続けて，

MERGE“EXAMPLE1”

のようにするのです。RUNすると，

INPUT FILE NAME :

と表示されますが，それにはリターンキーを押すだけで結構です。

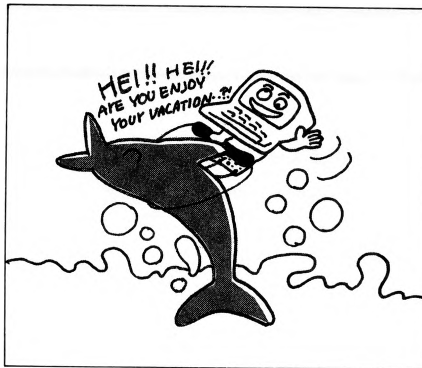
これで駄目ならプログラムに入力ミスがあるものと思われます。（影山　裕昭）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

**宛先：〒108 東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
「Oh! X質問箱」係**

STUDIO X ON AIR



FROM READERS TO THE EDITOR

もうすでに，外は夏真っ盛りかと思うほどの暑さ。でも，編集部の中はクーラーが効いているからすずしい，……はずなのだが調子が悪いのか，なんか暑い。仕事なんかできなーい。というわけにもいかず，しかたなく働くのであった。

◆本屋で「Oh!X」を手にしたとき，やっとディスク付録がついたかと思った。レジで「780円です」といわれて700円しか持っていなかった僕はスクーターをとばして80円を取りに戻った。ディスクを立ち上げて苦労しただけのことはあったと思った。　能美　和具(19)熊本県

すいませんねえ。しかし，予告で780円になりますと断ってあったとは思うんですが。5月号は立ち読みもしなかったのかな。

◆今月号は付録つきのためか異常に薄い！　と思って購入したら「Oh!PC」ではなく「Oh!X」でした。貴誌の迫力（？）ある誌面に洗脳され我が家配備第1号はPC-286シリーズのハズがX68000PROIIになりました。今後のご発展を祈ります（期待してます！）。

坂本　愼太郎(29)東京都

やった，偶然とはいえ読者が増えた。しかし，Oh!PCのほうでひとり減ったのかな。同じ会社の雑誌だからなあ。

◆半年に1回ぐらいはディスクをつけてもバチは当たらないと思う。　朝野　貴敦(17)滋賀県

いや，バチが当たって倒れる人が続出して本が出なくなるような気がします。

◆3週間に1度ぐらいディスクつきにしてほしい。　野田　佳照(16)愛知県

？？？？？？？？？？

◆今回のディスクは僕の人生に常駐した。

中島　潤史(15)埼玉県

早くワクチンを使って頭の中を治療しないとボロボロの人生になってしまうかも（どういう意味なんだ）。

◆創刊8周年なんですね。ちなみに私のX68000 PROは1歳です。だから，誕生日(買った日だよ)には1Mバイト増設してやりました。

山田　雅宏(18)岐阜県

このプレゼントで幸せになれるのはX68000か，あるいは本人か？　たぶん両方ともにでしょうね。

◆6月は創刊8周年記念だということを知って自分の誕生日も6月なのでうれしかった。

佐藤　直人(11)神奈川県

実は編集部のE.O.さんも6月に誕生日を迎えたので，みんなでケーキを食べました。ひさびさのケーキはおいしかったなあ。

◆今回のようなディスク配布が不可能であれば，ダンプリストを圧縮した形で載せてもらいたい。入力が楽である。　塩谷　隆治(31)広島県

そうですか。じゃあ，来月からリストを1文字＝0.1㎜ぐらいの大きさに圧縮して載せますので，皆さん虫眼鏡を用意して待っててください（冗談）。

◆いまYet Another Columnにはまっている。最初の頃は2時間ぐらい座りっぱなしだったが，いまでは自分の力量がわかってしまってか，2回ぐらいでやめてしまう。編集部には4万点をこすつわものがいるそうだが，どんな手を使っているのでしょうか。正当なやり方で取れるはずはないのだから，と思う。柴崎　誠(17)福島県

正当なやり方ですよー。どんなやり方かというと……。

◆編集部での40,860点とゆーのは信じられない。おそらく，「必殺！　ESCキーで止めればどこに落とせばいいかわかるじゃないか攻撃！」を使ったのだろーと思う。ぢゃなければ，古代フェニキアに生きた者の血を引一てるとか。

大島　貴成(17)栃木県

実はですねー，あのゲームには「なるべく高いところから落としたほうが点が高いんやでー攻撃！」というものがあるのですよ。気づいた人もいるようですが。

◆Yet Another Columnは面白かった。ブロックがくずれる音が「ケセランパサラン」と聞こえてしまうのは僕だけだろうか。そして，このゲームを「ケセランパサラン」と呼んでいる。友達におかしいといわれたが，なにがなんでも「ケセランパサラン」と呼んでいる。

奥村　真明(17)埼玉県

僕も自分のイスを「ギシギシくん」と呼んでかわいがっています。また，その友達には時計の「コチコチくん」というのがいます。

◆やられてしまった。かねてから作ろうと思っていた「この木なんの木」。日立社員のこの私が作ろうとしたのに会社が忙しくて……。P.S.習志野工場のお昼にはかかりません。さみしいなあ。ほかでは確かにかかるところもあるのに。ところで，一般人には「この木なんの木」が日立の社歌だと思っている人がいるようだが，これは社歌ではありません。社歌は別にあります。念のため。　藤井　実(19)千葉県

アンケートハガキを読んでいても「この木なんの木」がよかったというのが予想外に（？）多くてびっくりしました。別に「よかった」というのが予想外なのではなく数が本当に多くて。

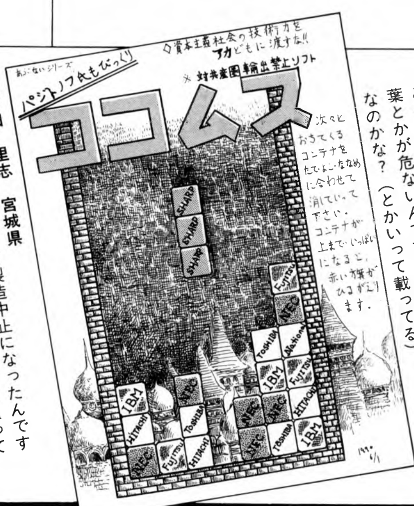
◆Oh!Xを買いはじめて7，8カ月。そろそろ内容のペースにもついていけるようになりました。最初は内容についていけず（突然OPMAだとか書いてあったので）とても困り，なんて不親切な本なんだろうと思いましたが私もいいかげんなもので，いまではもっといろんなツール載せろーと心の中では思っています。これからもいろいろなプログラムを載せてください。次回の付録ディスクが近いうちにあることを願っております。では，さようなら。

小田　典央(19)静岡県

まあ，人間ってそういうもんですよ。



◀吉田　里志　宮城県
えっ，パックマニアって製造中止になったんですか。本当ですか。ところで，あれとイースIIIって全然違うんですが，忘れられるんでしょうか。



◀大野　真実　静岡県
あぶないシリーズ第2弾です。しかし，本当に言葉とかが危ないんですけど……。載せても大丈夫なのかな？（とかいって載ってる）

◆もっと本を厚くして5月号に載っていたX68000の変なデモみたいのをたくさん載せてもらいたい。　　　　　　　　望月　伸幸(17)静岡県

変なデモというのはひょっとして（で）のショートプロぱーてぃに出ていた例のアレのことでしょうか。本が厚くなってああいうのばっかり載っていたらとてつもなく恐ろしいような気がしますが。

◆うーん，なにか押し入れの中でカサカサ音がするなあ……，と思って押し入れ開けてゴソゴソやっていたら，「ゲッ！」，思っていたとおりゴ，キ，ブ，リとご対面してしまった。予想していたこととはいえ，やっぱり気持ち悪い。と，躊躇していたらフトンの中に逃げ込まれてしまった。おそらく，まだ中にいると思われます。田舎にいた頃は東京近辺よりは湿気が少なかったせいか，ほとんどゴキブリは見たことがなかったのですが……。ゴキブリを見かけたせいで，「ああ，俺も関東に住んでいるんだなー」と妙に感激してしまった。が，やっぱりイヤだなー。でも，早めにやっつけんといかんな。うん。
工藤　隆(20)埼玉県

ゴキブリのもっとも恐ろしいところとは……。それはやはり，叩き殺そうとしたら顔に向かってバタバタバタと飛んでくるというところでしょう。あの瞬間のこわさときたらこの世で1番じゃあないかと思ったりします。

◆僕の友達が考えた“パソコンとカツ丼を手に入れる方法”。

1) 展示パソコンを持って逃げる
2) 逃げる途中に隠れている友達にパソコンを渡す
3) わざと警察に捕まる
4) 黙秘権を使う
5) しばらくすると警察がカツ丼をくれる
6) もうちょっとすると釈放される

これでパソコンが手に入りカツ丼も食える。すごい！　　　　　小川　伸一郎(15)京都府

いやー，すごいですね。15歳（？）にしてこの頭脳。編集部一同思わず感心してしまいました。まさに完全犯罪ですね。ひょっとしてノーベル賞もらえるかも。どうもおめでとうございます。

◆X68000も10万台をこえたようなので，そろそろマニア以外にも売ることを考えたらどうだろう。案としては自己診断機能の高度化。たとえば，まずコンセントを入れると周辺機器をチェックする。ディスプレイやキーボード，マウスが接続されていないと，「私の顔をつけて」とか「私のねずみはどこ」と話して誰でも接続できるようにする。　　　笠井　康彦(23)神奈川県

すると，接続を間違えたりすると「そこじゃないわよ」とか，スイッチを切ろうとすると「やめて」とかしゃべるんだろうか。あー，気持ち悪い。

◆PC-9801と同じくらい普及しているビジネスパソコンであるといってX68000を買ってもらったのに例のウイルス事件によってうそがばれた。　　　　　　　　森下　剛(14)京都府

そんなすぐにばれるようなうそを……。



▲杉本　秀昭　宮城県
なんかよくわからないけど，ともかくそんな格好してると風邪をひきますよー。とかなんとか，いらんこといったりして。



▲大村　直人　北海道
どうもありがとうございます。やはり，こういうお祝いのハガキもないとね。しかし，とっても味わいのある(?)イラストですね。

◆最近，アクションゲームやロールプレイングゲームに興味がわかなくなった。どうしてだろう。　　　　　　　　中井　卓(18)大阪府

どうしてだろう。きっと大人になったんだよ。

◆涙の浪人生活に入ってから小遣いを1,000円に減らされてしまった。しょうがないので弁当を作ってもらえなかった日に食事をぬいて300円ほどひねりだし，やっとOh!Xと好きなバイクの雑誌を買っている有り様。なんとも情けないことであります。しかたないですけどね。最近はゲームもあんまりしていなかったので（というより，「これ！」と思うのがなかった）ポピュラスを知ったとき，はまってしまいそうでこわいと思いながら金がないのでさみしく思っておりました。そこにこのプレゼント。僕にポピュラスをくれー。Oh!Xを買い始めて7年目。小学生だった僕もいまは浪人生，なんかすごいものを感じるなあー。　　　　　安陪　亘(18)三重県

ううっ，なんて情けない。ごはんを抜いてその浮いたお金でなにかを買うというのはよくある話ですが，体をこわさない程度にしましょう。でも，そうかといってポピュラスをあげるわけにはいかない。

◆ふと思った。ファジィコンピュータ内臓（ぢゃなくて内蔵）のカメラで撮った写真はどのようになるのか。
fuzzy(形)「中略」　2．[写真が]ぼやけた(blurred)　－シニア英和辞典　4訂版より－
大村　直人(17)北海道

なるほど。

◆暑さが厳しくなってきているなか，部屋に閉じこもりっぱなしだと頭がどうにかなりそうです。懸賞にクーラーもつけてください。
荻久保　雅道(14)静岡県

僕もクーラー欲しい。

◆以前，続けて4回足を運んだ映画のサントラ盤をステレオを持っていないのに買ってしまって，そのレコードのためにステレオを買ったことがありました。今月号の付録のディスクを見てふと思い出してしまいました。
三原　克之(36)福岡県

そういえば，僕もCDラジカセしか持ってないのにレーザーディスクのソフトやレコードを持っている。

◆X68000が10万台前後だそうですが，もし個人でソフトハウスを開業したとして1パーセント以上の人が（通信販売で）ソフトを購入すれば経営が成り立つと思います。「私はやってみたい！」と思っている人はかなりいるのではないでしょうか。ですから，ソフトハウス経営についての特集をお願いします。特に，ダビング工場のメーカー名と連絡先やその手数料，パッケージの単価と依頼数量など。この特集をすることにより，X68000ユーザーの中からソフトハウスを開業する人が多く出る→ソフトが増える→X68000購入者が増える→Oh!X購入者が増える！　　　　　　高久　裕明(29)東京都

やはり，問題はその個人が作ったソフトが市販ソフトとして受け入れられるようなレベルに達しているかどうかでしょう。つまらなければ，やっぱり全然売れないだろうし，面白ければ販売しようという話はどこかから来るでしょうから。

◆ANGELの人体モデルはどうして女の人なのですか。　　　　　竹永　昌伸(16)兵庫県

うっ，それだけは聞かないで。じゃなくて，ただ単に男だと気持ち悪いからじゃないでしょうか。

◆アンケートハガキの何パーセントが読まれているのだろうか。読まれなければなにを書いても出さないのと同じだもんな……。
小杉　雅信(21)愛知県

全部読んでいるに決まっているじゃないですか。このコーナーやハミダシっていうのはアンケートハガキによって成り立っているんですから。だから，白紙とかでなくなんか面白いことを書いて出してください。スタッフの人なんかもくるたびにハガキを読んでますよ。

◆いつもOh!Xの記事を見て，すごくうらやましくなります。なぜかといえば，SHIFT BREAKとかmicroOdysseyとかみたいに自分の考えを自由に（多少は制限があるでしょうが）書けて，ま

たそれに対して読者から意見がきて，またそれに対して意見を言えるという。なんか，そういうのっていいですよね。いちばんうらやましいのはやっぱり「STUDIO X」の答える人かな。一度でいいから代わってほしいと思うのは僕ぐらいなものでしょうか。　斉藤　哲哉(18)愛知県

そんなにうらやましいですか？　まあ，一応仕事としてやっているんですが，確かに自由に書いたり，その反応が返ってくるというのは実に楽しいことです。

◆気がついたら，知らない人の家にいた。大学の芝生の上に寝ていた。梅田の映画館の中にいた。先輩，日本酒とビールのカクテルの中に味の素，塩，魚の頭，キャベツ，しょう油を入れて飲ませないでほしいなー（文科系サークルとは思えないところに入った……)。

佐藤　能久(19)大阪府

いや，体育系より文科系のほうが飲み会がきついというのはよくある話ですよね。しかし，魚の頭やキャベツだったらいいですよ。もっと，ひどい話を聞いたことがあります。それは，……（あまりにもひどくていえない)。

◆X68000のスーパーインポーズでうそのニュース速報（チャイムつき）を流し，バアさんを指名手配の犯人に仕立て上げたら，バアさん3日間悩んだ。　松本　浩一(24)栃木県

僕もそういうことを考えてPC-6601SRでやろうと思ったのですが，グラフィックが粗いので漢字がでかくなるし，第一，専用ディスプレイがなくてスーパーインポーズができなかったのでした。ううっ，悲しい思い出だなあ。

◆ゆるせないぜ！　アンケートハガキの下の“X68000（無印，ACE，PRO……)”の無印てのはなんだよー。初期型はなー，グラディウスが付いてたんだぞ。CZ-600C万歳！

御宿　桂治(18)山梨県

何をいってるんです。無印良品っていうじゃないですか。うーん，しょーもない答えになってしまった。

◆いま気がついたのですが，アンケートハガキの裏表に年齢を書く場所があるのには意味があるのだろうか（すでにどなたかが気づいているかもしれないが)。もしかして，裏の年齢は愛機の年齢を書くのだろうか。

西谷　健吾(17)兵庫県

違います。裏には数え年を書くんです（またまた，しょうもない答え)。

◆HDタイプのX68000は地震に弱いので対策を立てました。それはキャリングハンドルを利用して天井からロープで吊るすのです。そうすれば，ソバ屋の出前バイクの法則によりX68000は地球の重心に対して静止するのでクラッシュの魔の手から逃れることができます。ぜひ，おためしください。それにしても大洋は強い。

矢地　雄(18)東京都

部屋が広ければ問題はないけど，せまかったらロープの長さによっては悲惨なことになりそう。壁にぶつかって。そうでなくても，落ちたときのことを考えると，とてもおためしなんかできない。

◆なんということか。「ハード」のプレゼントがないじゃないか！　私は楽しみにしていたのに（当たるわけもないけど……)。今月号はX68000を持っていればとってもうれしいのかもしれないが，ほかのユーザーはどうしろっていうんだ。　秋友　謙二(16)山口県

「ハード」のプレゼントは今月だったんですよ。はっはっは。しかし，なかなか当たるのは難しいでしょうね。

◆読者の方に聞きたいんですけどマウス，トラックボール，みんなはどっちを使っているのでしょうか。私の場合，部屋が狭い（4畳半，バス，トイレ，キッチン共同で家賃8,000円。今春から1,500円上がった。くるしー)ので机の上にキーボードとサイバースティックを置くといっぱいになり，マウスとして使うスペースがなくトラックボールとして使っています。両手はふさがりますが，そのぶんマウスのときのような腕の筋肉痛（あるわけねー）がなくなります（運動量が少ない)。みなさんはどっちです。

栗　幸司(21)広島県

僕はマウスとして使っていますが，机の上の空きスペースが10×10cmぐらいしかないので非常に苦しい。

◆初のフロッピーディスクの付録，年寄りには最高のオマケでした。長いリストを打ち込むことは体力が持ちません。最近はリストを見るだけであきらめていたものでした。年寄りのためにもこれからもときどき入れてほしいと思います。　小池　清(42)滋賀県

年寄りというほどの年でもないと思うんですが。まあ，長いリストを打ち込むのってけっこう体力が必要ですもんね。

◆愛読者年間モニタの応募者が欠員というのは，とても残念です。読者の皆さんがどうせなれないだろうと敬遠しているのか，本当に参加意識が薄れているのかはわかりませんが，7名というのには驚きです。私は第1期のモニタをさせていただいたので前者のほうですが，モニタ経験のある者でももう一度できるものならぜひやりたいところです。たぶんあの記事に刺激されていまではかなりの数の応募があると思いますが……。　紺谷　憲児(22)大阪府

別に一度やったからといって，年間モニタが二度とできないということはありませんから，経験者の方もどんどん応募してきてください。

◆1年ぶりにX1turboと再会した。が，2,3回スペースキーを叩くとスペースキーが死んだ。こうなるとほとんどのゲームができない。しょうがないのでワープロとして無理に使っていた。でもこれでは面白くないので，近くの電器屋に修理に出したらキーのスイッチとカールコードの交換で1万円以上もした。おかげで翌日のビジネスショウに行けなくなった。しかもである。スーパー大戦略をやっていて気がついたのだが，HELPキーが死んでいる。どーしよう。あんまり使うキーでないだけに悩んでしまう。

加藤　健二(18)埼玉県

まさに「一難去って，また一難」。

◆やっぱりX68000はいいですね。あっ，そういえば4月のいくんちだったか忘れましたが，夜，MOTOSを立ち上げたらいつものオープニングの曲と違う曲が流れたんです。あれは，なんだったんでしょう。　野口　智広(17)神奈川県

さあ，なんだったんでしょう。

◆バットモービル届きました。こんな凄いプレゼント生まれて初めてです(笑)。とりあえずディスプレイの上に飾ってあります。暇になると走らせてみたりしていますが，傍から見るとちょっとあぶないやつに見えるかも（かもじゃないって)。　松久　孝治(20)岐阜県

走らせるときに「ブーン，ブーン」とかいってやると，なかなかいいかもしれない（なにがいいのやら……)。

◆はじめまして。僕はX68000を買って（もらって）1年と少したちました。買って1カ月ほどたってから今まで，BASICを興味だけで学んできました。自分ではなかなか進歩したと思って，そろそろ高レベルの雑誌を購入しようと思いOh!Xを買うにいたったのです。が！　内容を見たとたん，全身の血が凍ったかと思うほどにおどろいた(なんじゃそりゃ)。今まで僕がコツコ

◀清水　健年　東京都
こんな本が出たら売れるんだろうか。うーん，やっぱり結構売れてしまうんだろうなあ。しかし，読んだら飲んでみたくなるんだろうなあ。

◀信川　洋　東京都
ポピュラスって本当に人気が高いんですよね。シナリオディスクのプロミストランドも出たことだし，ますますハマル人が続出するのかな。

ツ学んできたことは，まるで宇宙の中の人間……まではいかないが，星のような（あんまし変わらん）ものだったのです。そんで，今月号の中に「PROを買って半年たって……」などといいながら僕にはわけのわからんことが書いてあったりします。いったいこーゆー人はどうやって学んだのか……。教えて！

田村　高志(16)愛知県

人は人，自分は自分ですから，マイペースでコツコツやるのがいいんじゃないですか。試行錯誤しながら自分で学んだほうが身につくし，面白いですからね。

あとがき

ここに載っているのは6月号のハガキからなんですが，6月号のアンケートハガキはやはり，ディスクに関するのが多かったですね。目についたものをちょっとまとめてみました。

・Yet Another Columnが面白くてはまった（定期試験があるにも関わらずとか，忙しいのにとかいうのが多い）。

ちなみにハガキで書いてきた中でのハイスコアは滋賀県にお住まいの小野さんの50,875点でした。

・ディスクつきになるんなら，○○円出しても買う。

・ディスクをフォーマットしてしまった。

こういう人が何人かいましたが，やはりプロテクトシールを貼っておいたほうがよかったのかな……。まあ，そういう場合はしょうがないのでそのディスクをこちらへ送ってもらえればもう1回書き込んで送り返してあげられると思います。

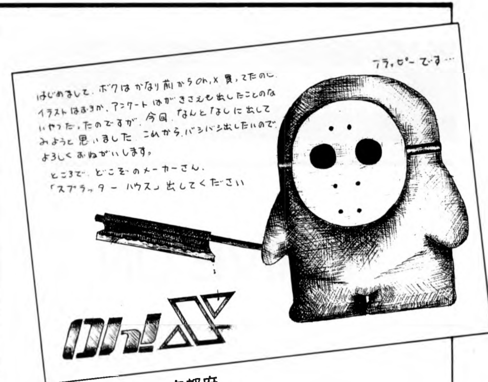
・6月号は保存版にしたいので2冊買った。

・ANGELが走らない。

ANGELはまったく走らないと思っている人が多いようですが，そんなことはありません。

勘違いをしている人がいるかもしれませんが，6月号の49ページの“2) FLOAT2+.Xの動作がおかしくなることがある（たまに計算を間違う)。ANGELは実行しないように。”というのはFLOAT2+.Xを使ってANGELを実行しないようにという意味で，ANGEL自体がおかしいということではありません。FLOAT2.Xを使えばANGELはちゃんと動きます（ANGELには多少おかしなところもありますが，起動は絶対にできるはずです)。走らない人はもう一度本文をよく読んでください。

なんか違うコーナーのようになってしまいましたが，今月は大部分がディスクに関するハガキでしたので，あしからず。



▲熊谷　逸郎　京都府
スプラッターハウスに……。なんでフラッピーがスプラッターハウスに……。でも，なんか似合ってますよね。マスクを取ったときに爆笑されそうだけど。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。

仲間

★「パ○レイバー」のサークル「狂走都市」ではCGを大々的に扱っている会報を発行中です。CGのPDS発行も企画しています。「パ○レイバー」だけでなくほかのゆうき先生の作品をCGで描いてみたい15歳以上の方，入会してください(当会所有機種はX68000 EXPERT，PC-9801RX21)。62円切手同封でご連絡ください。〒949-66　新潟県南魚沼郡六日町大字六日町484　種村聡人(15)

★X68000ユーザーを対象とした（X1でも可）サークル「X68K-GR」の会員を募集しています。ゲームを中心に月1回程度の会合を開き，ゲームの情報交換や会誌の発行をしたいと思います。興味のある方は簡単な自己PRと62円切手同封のうえ，下記の住所までご連絡を。〒239　神奈川県横須賀市ハイランド5-3-19　三浦正義(17)

★「ディスクサービスX68000」。このサークルではX68000の機能をより高度に活用するためにユーザーの方を募集しております。活動内容は皆さんから集めたプログラムを会報にまとめ，配布するというものです。興味のある方は62円切手を同封のうえ，ご連絡ください。〒165　東京都中野区上鷺宮5-28-24　前野千絵(21)

★このたびサークル「ぺけろく亭」では会員増強に伴い第3期会員を限定募集いたします。パソコンをより高度に活用するための情報交換などが主体です。入会希望の方は62円切手同封のうえ，下記までご連絡ください。〒491-03　愛知県一宮市萩原町富田方字茶原54　野杁真広(23)

★このたび，X68000ユーザーのサークルを作るにあたりましてプログラマを募集しようと思います。できればアセンブラやC言語のできる方，興味があってこんなボクでも力になれるなら……とか思った人はぜひご連絡を。〒437-11　静岡県磐田郡浅羽町浅羽1169-32　袴田信孝(17)

売ります

★MZ-2000用周辺機器を以下の価格で（送料込み，値引き可)。フロッピーディスクドライブ「MZ-1F07」を2万5千円，プリンタ「MZ-80P6」を1万円。いずれもインタフェイス，ケーブル，説明書つき。拡張ユニット(MZ-1U01)，漢字ROMボード(MZ-1R13同等品)，VRAM 3ページを各8千円で。連絡はハガキで。〒371　群馬県前橋市上小出町6-1-205　中嶋康弘(31)

★X1用データレコーダ「CZ-8RL1」を5千円（送料込み)で。連絡は往復ハガキで。〒737　広島県呉市阿賀中央1-24-20　清水幹雄(18)

★MIDI音源ローランド「D-110」(1年使用）をマニュアル，保証書，付属品一式，1Uラックつきで4万5千円で譲ります。送料当方負担。連絡は往復ハガキで。気長に待ちます。〒594　大阪府和泉市鶴山台4-8-3　山路智弘(20)

★IO-735X（カラーイメージジェット)，マニュアル，付属品，X68000用ケーブルつき，箱あり，キズなしを送料込み8万円で。連絡は往復ハガキで。〒238　神奈川県横須賀市深田台76　小林秀樹(21)

★ワープロ書院「WD-540」を5万円で。箱，マニュアル，付属品，保証書あり。おまけでインクリボン，フロッピーディスクをつけます。連絡は往復ハガキで。〒187　東京都小平市花小金井4-286　小川和幸(24)

★増設RAMボード「CZ-6BE1」を送料込みで2万2千円程度で。少しぐらいなら割引します。マニュアル，付属品あり，キズなし。連絡はハガキか手紙でお願いします。〒665　兵庫県宝塚市南ひばりが丘3-26-5　関口敬文(15)

買います

★X1用FM音源ボード「CZ-8BS1」(完動，箱，マニュアル，付属品あり）を送料込みで8千円から1万円で。連絡は往復ハガキで。〒673　兵庫県明石市西明石北町3-16-5　春名隆行(17)

★MZ-2521用辞書ROM，増設RAMボード，増設VRAMボードを各6千円ぐらいで。コンパチ品も可です。希望価格を明記してハガキで。〒458　愛知県名古屋市緑区鳴海町神明163-1　安川実(16)

★X68000用増設RAM「CZ-BE1」を1万5千円くらいでお願いします。完動で付属品つき。連絡はハガキで。〒285　千葉県佐倉市井野869-26　松本琢磨(17)

バックナンバー

★Oh!X1989年5月号を千円，「X68000テクニカルデータブック」を2千円で。美品希望。連絡はハガキで。〒289-13　千葉県山武郡成東町成東2470　安井忍(22)

▶最近このコーナーで紹介しているサークルに対して，問い合わせても返事がこない，他人の著作権を侵害する行為が行われているなどの残念な報告が増えています。信頼関係に基づくコーナーですからマナーの向上に留意してもらいたいものです。今後，仲間のコーナーでの紹介を希望されるサークルは必ず会誌の見本を送ってください。

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今回は6月号の記事に関するレポートです。これまでのモニタの方にとっては最後のレポートです。1年間のモニタレポートご苦労さまでした。

●“共通システム”という考えに基づいたS-OS思想は，5年たったいまでも決して古びてしまったものではない（それどころか，いまだからこそ重要なことかもしれない）。しかし，“SWORD”というシステム自体は8ビット全盛時代のものでしかない。16ビット以上の時代の8ビットのためのシステム，“Excalibur”なり“Storm Bringer”なりを発表すべきではないだろうか。INTEGRAL XI (KAME-DOS)のように，どんなディスクフォーマットも読める機能やヒストリといった近代的機能を備えるS-OSが発表されてもよいと思う。だからといって，テープユーザーや旧機種ユーザーを切ってもいいというわけではないが。
西田宗千佳(18) X68000, X1Fmodel20 千葉県

●なんといっても“PurePASCAL”，これに尽きるのではないでしょうか。グラフィック機能が標準でないのが残念ですが，これが付けばかなりなものになると思います。私のようなPASCAL派はX68000ユーザーの中には少ないかもしれませんが，これを機にPASCALを勉強してPASCALの素晴らしさを知ってほしいと思います。できれば，コンパイラの内部仕様などに関する記事があってもよかったのではないかと思います（まあ，これは今後に期待しましょう）。
森川一(24) X68000ACE-HD, XIturboII 北海道

●「ハードウェア工作入門」についてですが，製作の対象とする回路はできるだけシンプルなものをお願いしたいと思います。また，なるべくローコストでということも。最近は気軽にハンダゴテを握ったり，紙工作したり，プラモデルを作ったりというような話をあまり聞かなくなりました。これはやはり，身の周りに完成品があふれているためでしょう。でも，誰でもみんななにかしらの創作意欲を持っているはずです。「ハードウェア工作入門」には，そんな私たちの創作意欲を刺激し満足させてくれる連載になってほしいと願います。
藤田康一(19) X68000PRO 静岡県

●S-OSがまさかX68000やPC-286にまで広がるとは思っていませんでした。うれしいかぎりです。でも，X68000ユーザーはともかくとして，PCユーザーがこのことを知らないのは残念だと思います。Z80シミュレータとしてなんとかPCユーザーに知らせる方法はないでしょうか。ちなみにPC-9801RSで動かしてみましたけどX1よりも少々速いような気がしました。なんといっても2Dのディスクの読み書きができるのは5重丸です。
末吉克行(21) X1G, MZ-731, FM-7 兵庫県

●「ハードウェア工作入門」ですが，前回のアンケートで書いたことはちゃんと押さえてあり，「何が必要であるか」ということがわかりやすく書いてありました（さすがだなあ)。プログラムのように，簡単にはやり直しが効かないハードウェアが相手ですから，なかなか大変だと思います。入門講座の場合いちばん大切なのは,「急に難しくならない」ことだと思います。余談になりますが，NHK基礎英語がいまだに入門講座として利用されることが多いというのは「急に難しくなることがない」からなのだそうです。そうしないと，ついてこられないというわけです。バカ丁寧すぎるくらいでいいですから，ゆっくりのんびりやってほしいですね。それと，なるたけわかりやすい図を使ってほしいと思います。ジョイスティックポートにつなぐものがほとんどと聞いて安心しました。実際に組み立てる場合，回路図と配置図がパッとは結び付かないものです。毎回言っていることなのですが，「難しいことばは，脚注などを付けてもらいたい」と思います。X68000マシン語講座がなぜ読みやすいかというと，脚注などが詳しく，難易度に気を配っているからだといえます。ハードとソフトの違いはあるとはいえ，やっぱりこうあってほしいと思います。
湯澤聡(27) X68000, XIturboIII, MZ-2861/2531, PC-6601, MSX, PC-1360K 埼玉県

●アンケート結果を見て。やってくれますねー。まあ，そうとうの内輪ネタであるのですけれど，こういった内容であればいたしかたないでしょう。さすがのX68000の伸びと，ほかの項目内のX68000が占める割合が，いまいちばん私にとってショックですね。あのとき，XIturboZ か X68000か多少なりとも悩んだんですから。ベストライター（もちろんOh!Xのスタッフ1人ひとりはベストライターです)，なんていうアンケートはまさか載せるためだとは思いませんでした。祝一平氏がトップなのは，やはりという感じ。ま，ほかにもいろいろありましたが世論調査みたいでいいですね。作り手と受け手がこうもコミュニケーションできるのはOh!Xだけでしょう。また，やりましょう。
大津和之(20) XIturboZ 福岡県

ごめんなさいのコーナー

7月号 AFTER REVIEW

1 「サーク」のレビュー内の写真が「ルーンワース 黒衣の貴公子」のものと入れ替わっていました。関係各位にはご迷惑をかけました。お詫びいたします。

7月号 WZD

先月号のものではコマンドラインからパラメータ付きで実行した場合，復帰時の動作が保証できません。詳しくは今月号のP.147をご覧ください。アセンブル時は問題なく動作するはずです。

6月号 ANGEL

P.65 回転のコマンドの書式に間違いがありました。

rotx 〈式〉 → rot.x (〈式〉)
roty 〈式〉 → rot.y (〈式〉)
rotz 〈式〉 → rot.z (〈式〉)

のように変更してください。お詫びして訂正いたします。

また，画面をはみだすような絵を描かせると止まってしまうことがあるようです。

6月号 GCC Ver.1.36.01

Humanのバージョンが2.00の人はうまく動かないようです。前にも書いたようにHuman v.2.00はシャープでv.2.01に交換してくれますので，これを機にバージョンアップしましょう。

6月号 X68000マシン語プログラミング

files.hのリストが抜けていました。詳しくは今月のX68000マシン語プログラミングをご覧ください。申し訳ありませんでした。

バグに関するお問い合わせは
☎03(5488)1311(直通)
月〜金曜日 16：00〜18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

本誌創刊号をプレゼント！そんなバカな～っ

▼お陰さまで本誌は通巻100号を迎えることができました。今月号は，グラフィック特集，表紙ぎゃらりぃなどでカラーページを増ページして豪華にお送りいたしましたがいかがでしたでしょうか。

さて，編集部ではこれを機に取り置きのバックナンバーを整理し，その一部をなんらかのかたちで皆さんに提供したいと考えています。とりあえず今回は100号記念プレゼント番外編として，Oh!MZの創刊号を3名の方に差し上げたいと思います。ご希望の方は綴じ込みのアンケートハガキのプレゼントNo.に0と記入してお送りください。

▼本誌では，コンピュータサークルなどの制作による同人ソフトの紹介を考えています。特にX68000などのユーザーグループの作品にはレベルの高いものが多く，市販ソフトにはない手作りの味が魅力です。これらはパソケットなどを通じて安価に販売されていますが，一般にはあまり流通していません。本誌ではこうした作品を広く読者の皆さんに知ってもらいたいと思います。本誌での紹介を希望するソフトがありましたら，編集部までご連絡ください（☎03-5488-1309)。また，団体名，連絡先，代表者名を明記のうえサンプルソフトをお送りいただければ幸いです。

▼ここで嬉しいお知らせです。しばらく本誌を離れていた清水和人氏が次号より復帰。ゲームやプログラミングの楽しい記事をお願いすることになりました。ご期待ください。

▼先月号でお知らせしたとおり，7月1日から株式会社日本ソフトバンクは「ソフトバンク株式会社」と社名を変更しております。また，社屋も移転となり，Oh!X編集部は16日より新しい編集部にて業務を開始しております。お問い合わせの際には，電話番号が変わっておりますのでご注意ください。

▼先月号に掲載した日コン連企画㈱の広告中，X1ユーザーに対して不適切な表現があり，ご迷惑をおかけしました。広告主になり代わり，深くお詫び申し上げます。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㋢㋼㋮㊔」係

S H I F T · B R E A K

▶マーク・トウェインの『不思議な少年』を読んだ。天使“サタン”が人間の矮小さ卑俗さを描いてみせる。そのなかに彼が泥の小人を箱庭に生活させ，城を作ったところで雷で皆殺しにするくだりがあった。彼は泥人間の運命なんか気にもしていない。なんて残酷な奴だ。……さて，読書はここまでにしてポピュラスでもやろーかな。今日は272面だ。(H.U.)

▶5月号でレビューした天下統一だが，実はとても速かったのだ！　愚かにも「もろCPUの速さがでちゃうんだよう」などと書いたが，なんのなんの，製品版は80286のRXにも決してひけをとらない。いや，それ以上だといえる！　こんなところでフォローしても何人の人が読むかわからないが，とにかく！　おもしろい！(亀)

▶出張でスペインとポルトガルに行くことになったのだが，出発予定日の3日前になっても飛行機が何時に成田を出るのか知らされていない。それどころか，旅行会社は昼出発と言い，航空会社は夜出発だと言い張る。おまけに，旅行会社から聞いたポルトガルのホテルの住所は実在しない地名だと言う噂だ。果たして無事たどり着くのだろうか。(K.M.)

▶ポピュラス全500面クリア達成しました！　でも，エンディングのようなものはなく，0面（GENESISのようなもの）に戻ってしまうんですね。もしかして，2周しないとエンディングが見れないとか!?（R-TYPEみたいだな）今度は「対戦」と「プロミストランド」を究めてみようかと思います。Uさんこの間の勝負は練習ですよ，フフフ。(善)

▶ちょっとお尋ねしますが，皆さんの中でトマトジュースのお好きな方はいらっしゃいます？　僕はあれが好物なのです（野菜ジュースはもっと好き）が，友人から毒物飲料のごとくいわれてしまいました。健康飲料と称した妙なのがいっぱい出てくるずっと前からあった，由緒正しい飲み物のはずなのに（関係ないか)。誰か賛同を！(A.T.)

▶以前自分が担当してたころの質問箱のページを読み返してみたら，上手く書けてんだよ，これが。内容，文章ともに完璧に近い。俺って凄かったんだなーって本気で思ったね。つまずいたり，水たまりに落ちたり，犬の○○踏んづけたりするのを気にせずに空を見上げて歩いてるほうが気持ちいい。ということらしい（おや，雨だ)。(Mu)

▶なれど，高校生の頃から大嫌いだったのが“自然保護”という言葉であった。だって，保護というのは「自分より弱いものを守ること」ではないか。いつから人間は自然を保護できるほど偉くなったんだ？　いつからそんなに傲慢になったんだ？　もっと謙虚になりなさい。謙虚に。そして，正直に「人間保護」とでもいってなさい。私は悲しい。(K)

▶ネット上のジョークを真に受けるヤツ。7月を待たずにウイルス終結宣言をするお役所。しかし，ウイルス学会というのはウイルスを作ろうという学会だったとは……。さて，POPULASの決め手は序盤。2つ目の城を何秒で作るか，いかに海を制し，どれだけ速く侵略できるかにかかっている，と思う。マップのせいにしちゃいけませんよ。(S.N.)

▶新社屋となるNS（日本ソフトバンクではなく，日本食堂の略）高輪ビルへ見学に。下には富士銀行，上にはレストランという結構な趣。ところでその日は変な考えばかり浮かぶ日で，社長室を見ては「6万円ぐらいで貸してくれないかな」とか，帰りにNECのスーパータワー（風穴のあいたビル）を見て「あっ，クレイジークレイマー」とか……。(A)

▶自慢じゃないがシリーズその2。私はいわゆる霊現象によくあう。台所に鎧武者が出たり，遊体離脱や寝入り端の子守歌なんてのはザラ。予知夢も多いし，デジャヴってやつも日に3回くらいある。ほら，こうやって原稿を書いてるのだってあった気が……。そういや昨日にも，その前にも……，ん？　そりゃ単なる習慣だって。(E.O.)

▶創刊100号。私が編集に加わってから52冊，半分以上になるのか。半年前まで最若手だったのに……。編集部が大使館立ち並ぶ千代田区からお寺の並ぶ泉岳寺へ移転することになった。思えばここも3年半。さらば，武道館，靖国神社，北の丸公園……テキ屋にダフ屋の群れ，50mごとに並んだ警官の列……。さらば白百合学園のセーラー服。(U)

◀おかげさまで6月号は売り切れ店続出，なかには500冊以上売っていただいた書店もある。1年間はバックナンバーが買えるよう在庫を増やしたのだが……。さて，その6月号にゲーム基板の話があったが，文脈上Oh!FMのY氏が基板評価に関わっていると誤解を招く部分があり，Y氏には申し訳ないことをした。この場を借りてお詫びしたい。(T)

microOdyssey

東京オリンピックで日本がアルゼンチンに勝ったときは，まだ私にはサッカーのなんたるかがわからなかった。目覚めは2年後のワールドカップ・イングランド大会の決勝で，地元イングランドと西ドイツが同点で延長戦に入り，結局イングランドが4－2で勝ったときだ。勝ち越しの1点は，バーに当たって落下し，ボールは外に跳ねかえったが判定はゴールであった。

抗議する西ドイツ選手たちを静めたのはキャプテンのウヴェ・ゼーラー。次のメキシコ大会の準々決勝で再びイングランドにリードされたが，なんとロスタイムにゼーラーはゴールに背を向けたままヘディングシュートを決めたのだ。準決勝はさらに激しいイタリア戦。肩を脱臼したベッケンバウアーがギプスで腕を固定してプレーを続ける姿は子供心に焼きついている。西ドイツはやはり終了間際に同点，延長で逆転。が，再度逆転され，さらに追いつくという歴史に残る死闘の末に敗れた。以来，私はずっと西ドイツの熱狂的（？）ファンを自称している。

ああ，それにひきかえ，なんでこんなに弱いんだろうと悲しくなるのが日本のサッカーだ。日本よりも弱い国なんて世界中さがしてもそんなにはない。それも競技人口からいえば結構大国に属するわけで，「いや日本じゃあまり盛んじゃないから……」と言い訳もきかないのだ。

日本が弱い理由は，1）技術がない。2）体力がない。3）センスがない。の3点が基本だが，もっと深い部分，思想的な面で問題があるような気がする。サッカーだけでなく。チームプレーを必要とする球技は基本的にダメなのだ。

たとえば，子供たちのサッカーで，キープ力のある子がドリブルで突破しようとすると，その子だけが「1人でやっちゃダメでしょう」と注意を受ける。周りの子供が注意されることは意外と少ないものだ。チームプレーに「力を合わせて，助けあい」というイメージが植えつけられるのはこのときからではないか。

プロのサッカー選手が誰かにパスを出すのは，それが自分にとって「もっともいいプレー」となる場合だ。逆にボールを持たない選手はパスをもらえる状況を作るのが仕事の基本だ。

ボールを持つ選手Aはできれば自力で突破したいし，その自信もある。だが別の選手Bが，いやオレにつなぐのがお前にとってのベストチョイスだといわんばかりに動く。Aは，しかたがない，いったん任せるがリターンをよこしたほうが身のためだぞ，と前に進む。この駆け引きの結果が真のチームプレーとなる。パスは助け合いではなく仕事なのだ。

また，精神面でも多くの教育的指導は勝負に向いていない。たとえば，日本人が大切にしている根性とか精神力とかいう言葉。なぜか「倒れるまで頑張る」ことを美徳と誤解している人が多い。玉砕しても負けは負けなのに，である。一方，西洋の精神力は「最後まで倒れない」ことをさす。なぜなら彼らは勝つために頑張るのだ。この違いは大きい。

ひいきの西ドイツは，今回のワールドカップで82年，86年に続いて決勝に進んだ。だが，これぞゲルマン魂という，逆境を勝ち抜く試合が今大会ではまだ見られない。それだけに決勝戦は波乱に満ちた展開を期待しよう。いまは決勝戦を2日前に控えた7月7日である。（T）

1990年9月号8月18日(土)発売

特集1　日本語を処理するために
特集2　2Dグラフィック続論

X68000にハンディスキャナをつなぐ

新製品紹介　ビデオボード/C compiler Ver.2.0(？)

新連載　清水和人流プログラミング道場/荻窪圭「大人のためのX68000」

全機種共通システム
ビリヤードゲーム

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(233)3312
	〃	書泉ブックマートB1	03(294)0011
	〃	書泉グランデ5F	03(295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(257)2660
	八重洲	八重洲ブックセンター3F	03(281)1811
	新宿	紀伊国屋書店本店	03(354)0131
	高田馬場	未来堂書店	03(200)9185
	渋谷	大盛堂書店	03(463)0511
	池袋	リブロ池袋店	03(981)0111
	〃	西武百貨店9F コンピュータ・フォーラム	03(981)0111
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期講読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で講読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期講読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(238)0700


8月号
■1990年8月1日発行　定価560円(本体544円)
■発行人　孫　正義
■編集人　橋本五郎
■発売元　ソフトバンク株式会社
■出版事業部　〒108 東京都港区高輪2-19-13　NS高輪ビル
Oh!X編集部　☎03(5488)1309
出版営業部　☎03(5488)1360 FAX 03(5488)1364
広告センター　☎03(297)0181
■印　刷　凸版印刷株式会社

©1990 SOFTBANK CORP. 雑誌 02179-8 本誌からの無断転載を禁じます。
落丁・乱丁の場合はお取り替えいたします。



料金受取人払

高輪局承認

1459

差出有効期間
平成4年7月
15日まで

郵便はがき

108-00

(受取人)

東京都港区高輪
2-19-13 NS高輪ビル

ソフトバンク株式会社

Oh!X 編集部行

電話

住所

氏名　　　年齢

職業・勤務先
学校・学部・学年

●編集部へのメッセージ

今月号の特集について

いちばん良かった記事

興味のなかった記事

これから載せてほしい記事内容

本誌以外にお読みのパソコン雑誌

推薦する市販ソフト

ソフト名：

推薦理由：

1カ月のおこづかいはいくらですか

あなたの愛機は（所有機種に○印をつけてください）　ない
X1（マニアタイプ,C,D,F,G,twin）X1turbo（model 10,20,30,40,II,III,Z,ZII,ZIII）
MZ-（80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861）
X68000（無印,ACE,PRO,PROII,EXPERT,EXPERTII,SUPER,HD）　その他
FD（　基）　TAPE　QD　HD（　MB）　プリンタ（　）

年齢　歳　パソコン歴　年　男・女　プレゼントNo.

振替用紙

←点線から、きれいに切り取ってご使用ねがいます。

払込票

通常払込料金
加入者負担

口座番号	東京 1 - 29307
加入者名	ソフトバンク株式会社
金額	億 千 百 十 万 千 百 十 円
払込人住所氏名	※
備考	受付局日付印

切り取らないで郵便局にお出しください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

払込通知票

通常払込料金
加入者負担

口座番号	東京 1 - 29307	金額	億 千 百 十 万 千 百 十 円
加入者名	ソフトバンク株式会社	料金	払込み / 特殊 円
払込人住所氏名	※ （郵便番号 　　　）	備考	
		受付局日付印	

各票の※印欄は、払込人において記載してください。

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。 （郵政省）

定期購読のお申し込みを頂きありがとうございます。この申込書の弊社到着〆切は次の通りです。これを過ぎますと次号からの発送となりますので、ご了承下さい。尚、郵便振替は郵便局で払い込まれてから当社に到着するのに、2週間位かかります。

Oh! PC	1日発売	前月 20日
	15日発売	当月 5日
月刊情報処理試験 BEEP メガドライブ		前月 末日
Oh! X Oh! FM THE COMPUTER C MAGAZINE パソコン マガジン		当月 10日

(御注意)

当月発売の号より前に遡ってからの申し込みは出来ません。

バックナンバーは、また別にお申し込み下さい。

切り取り線

切り取らないで郵便局にお出しください。

この欄は、加入者あての通信にお使いください。

送り先	お名前	フリガナ	性別 男・女	年齢	ご職業
	ご住所	フリガナ 〒			
	お電話	ご自宅	勤務先		

申込書	THE COMPUTER 定期購読	新規申し込み	継続申し込み TC NO.	年間 7,200円
	Oh!PC定期購読	新規申し込み	継続申し込み PC NO.	年間11,440円 6ヶ月 5,720円
	Oh!X 定期購読	新規申し込み	継続申し込み X NO.	年間 6,720円
	Oh!FM定期購読	新規申し込み	継続申し込み FM NO.	年間 6,720円
	月刊情報処理試験 定期購読	新規申し込み	継続申し込み JS NO.	年間 8,160円 6ヶ月 4,080円
	C MAGAZINE 定期購読	新規申し込み	継続申し込み CM NO.	年間11,760円
	パソコン マガジン 定期購読	新規申し込み	継続申し込み PM NO.	年間 6,960円
	Beep メガドライブ 定期購読	新規申し込み		年間 5,760円
通信欄				

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。 (郵政省)

まるごと新作ボリュームアップ号

POWERFUL MEGA-MAGAZINE

BEEP! MEGADRIVE

ビープ!メガドライブ

8月号 定価480円(税込) 好評発売中

創刊1周年記念
ご愛読感謝!
222名プレゼント



〈特別付録〉
ヘルファイアー
特製ポスター

僕達のまわりに異星人がいっぱい
コンピュータゲームの宇宙人侵略史を探る

何かと話題のUFO特集

新作ラッシュの秋を目前に今後の展開を予想する

メガドライブの'90年後半戦を占う

とじこみ保存版

フェリオス攻略ガイド

ヘルファイアー　スーパーモナコグランプリ
E SWAT　バットマン　四天明王

ゲームボーイ専門誌 パワーアップした第2弾だ!

ゲームボーイLIFE VOL.2

定価380円(税込)

- 54本のソフトを総ガイド
- 輝け!第1回ゲームボーイ大賞
- 試験にでないゲームボーイ講座

業界初の完全攻略 オールソーサリアンシリーズ

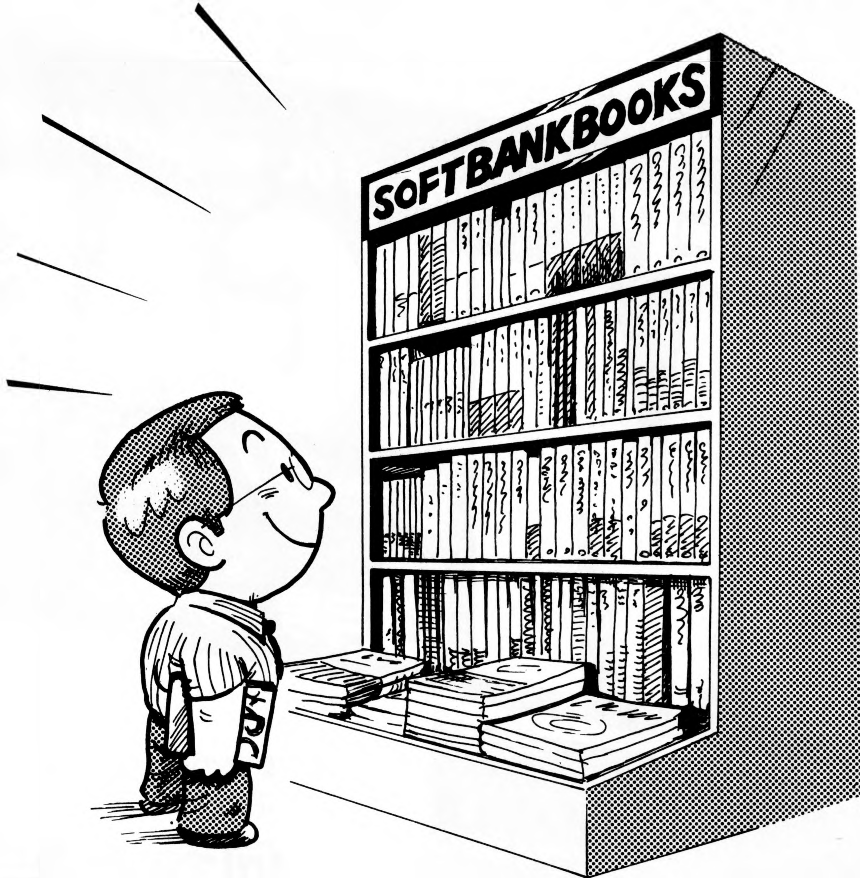
FALCOM MAGAZINE [ファルコム・マガジン]

定価680円(税込)

- オールアバウト・ソーサリアン パソコン版ソーサリアンの総ガイドに加え、メガドライブ版ソーサリアンも紹介
- オールファルコム・ベスト10 ファルコムユーザー100人が選んだファルコムなんでもベスト10

ソフトバンク

ソフトバンクの 書籍特約書店



下記の書店の一覧は、ソフトバンク書籍特約店として右にある商品の他、新刊もとりそろえております。ご希望の商品がある場合は、下記のお近くの書店にてお買い求め下さい。
(注)現品が売れて補充中の場合もございますので、ご注意下さい。



ソフトバンク出版事業部
〒108 東京都港区高輪2-19-13 ✆03(5488)1360

全国特約書店一覧

〈北海道〉		
札幌市	紀伊國屋書店札幌店	011-231-2131
〃	旭屋書店札幌店	011-241-3007
〃	丸善札幌支店	011-241-7252
〃	リーブルなにわ	011-221-3800
〃	冨貴堂札幌パルコ店	011-214-2303
〃	ダイヤ書房本店	011-712-2541
〃	ダイヤ書房西店	011-655-6223
旭川市	旭川冨貴堂	0166-26-3481
〃	ブックス平和マルカツ店	0166-23-6211
苫小牧市	旭屋書店苫小牧店	0144-36-5185
〈東北〉		
青森市	成田本店	0177-23-2431
〃	岡田書店	0177-23-1381
弘前市	紀伊國屋書店弘前店	0172-36-4511
〃	ブックイン城東	0172-28-2882
八戸市	伊吉書院	0178-44-1917
盛岡市	東山堂書店本店	0196-53-6464
〃	さわや書店	0196-53-4411
〃	第一書店	0196-53-3355
仙台市	金港堂	022-225-6521
〃	金港堂ブックセンター	022-223-0979
〃	アイエ書店駅前店	022-264-0718
〃	丸善仙台支店	022-266-1127
〃	髙山書店	022-263-1511
〃	ブックスみやぎ	022-267-4422
秋田市	三浦書店	0188-33-8131
山形市	八文字屋	0236-22-2150
福島市	岩瀬書店コルニエツタヤ店	0245-21-2101
〃	博向堂	0245-21-1161
郡山市	東北書店	0249-32-0379
いわき市	ヤマニ書房本店	0246-23-3481
〃	鹿島ブックセンター	0246-28-2222
会津若松市	宝文館	0242-27-5198
原町市	文芸堂	0244-22-1720
〈関東〉		
水戸市	川又書店駅前店	0292-31-0102
〃	ツルヤブックセンター	0292-25-2711
勝田市	武石書店	0292-73-1212
東海村	大野書店	0292-82-2098
鹿島郡	なみき書店	0299-96-1855
土浦市	共栄堂	0298-21-6134
つくば市	丸善筑波大学会館店	0298-51-6000
〃	友朋堂吾妻本店	0298-52-3665
宇都宮市	落合書店オリオン店	0286-34-3777
〃	落合書店東武ブックセンター	0286-34-8271
〃	新星堂宇都宮店	0286-33-2337
小山市	進駸堂駅ビル店	0285-25-1522
前橋市	煥乎堂	0272-23-1211
〃	リブロ前橋店	0272-34-1011
〃	戸田書店前橋店	0272-61-5063
高崎市	学陽書房	0273-23-4055
〃	サカ井書店	0273-62-1500
〃	新星堂高崎店	0273-27-3961
〃	戸田書店高崎店	0273-63-5110
太田市	ナカムラヤ	0276-22-2001
〈首都圏〉		
浦和市	須原屋本店	048-822-5321
浦和市	須原屋コルソ店	048-824-5321
大宮市	押田謙文堂	048-641-3141
〃	ブックセンター押田	048-647-3141
〃	三省堂ブックポート	048-646-2600
蕨市	須原屋蕨店	0484-44-1211
川口市	岩渕書店川口店	0482-52-2190
川越市	黒田書店川越店	0492-25-3138
所沢市	芳林堂所沢店	0429-25-5355
〃	いけだ書店所沢店	0429-28-3271
上福岡市	黒田書店上福岡店	0492-66-0120
朝霞市	文教堂朝霞店	0484-76-0107
志木市	新星堂志木店	0484-74-0182
春日部市	文教堂春日部店	048-752-7666
比企郡	錦竜サービス	0492-96-2962
千葉市	多田屋セントラルプラザ店	0472-24-1333
〃	キディランド千葉店	0472-25-2011
習志野市	厳翠堂	0474-72-5011
船橋市	ときわ書房本店	0474-24-0750
〃	リブロ船橋店	0474-25-0111
〃	旭屋書店船橋店	0474-24-7331
〃	芳林堂津田沼店	0474-78-3737
〃	第二厳翠堂	0474-65-0926
〃	三省堂書店西船橋店	0474-34-3111
柏市	西口アサノ	0471-44-2111
〃	新星堂柏店	0471-64-8551
松戸市	堀江良文堂	0473-65-5121
〃	辰正堂駅ビル店	0473-64-7997
横浜市	有隣堂トーヨー店	045-311-6265
〃	有隣堂東口ルミネ店	045-453-0811
〃	栄松堂相鉄ジョイナス店	045-321-6831
〃	そごうブックセンター	045-465-2111
〃	丸善ブックメイツポルタ店	045-453-6811
〃	有隣堂伊勢佐木店	045-261-1231
〃	有隣堂戸塚店	045-881-2661
〃	文華堂戸塚店	045-864-5151
〃	アーバン文華堂	045-821-5151
〃	文教堂青葉台南口店	045-983-5150
川崎市	有隣堂アゼリア店	044-245-1231
〃	有隣堂川崎BE店	044-200-6831
〃	文学堂本店	044-244-1251
〃	文教堂溝ノ口店	044-811-8258
鎌倉市	島森書店大船店	0467-46-3841
〃	鎌倉書店	0467-46-2619
横須賀市	平坂書房WALK店	0468-25-5537
藤沢市	有隣堂藤沢店	0466-26-1411
〃	リブロ藤沢店	0466-27-0111
〃	文教堂六会店	0466-82-9610
茅ヶ崎市	川上書店ルミネ店	0467-87-3827
平塚市	サクラ書店駅ビル店	0463-23-2751
〃	文教堂四之宮店	0463-54-2880
小田原市	八小堂書店	0465-22-7111
〃	伊勢治書店	0465-22-1366
〃	文教堂小田原店	0465-36-3677
厚木市	有隣堂厚木店	0462-23-4111
大和市	文教堂中央林間店	0462-75-4165
相模原市	文教堂相模大野店	0427-49-0650
〃	文教堂橋本店	0427-74-5581
相模原市	文教堂星ヶ丘店	0427-58-6121
津久井郡	文教堂城山店	0427-82-9278
〈東京〉		
千代田区	三省堂書店神田本店	03-233-3312
〃	書泉グランデ	03-295-0011
〃	東京堂書店	03-291-5181
〃	旭屋書店水道橋店	03-294-3781
〃	丸善お茶の水店	03-295-5581
〃	厳翠堂	03-291-1362
〃	いずみ神田南口店	03-254-8521
〃	明正堂秋葉原店	03-257-0758
〃	T-ZONE	03-257-2660
中央区	八重洲ブックセンター	03-281-1811
〃	日本橋丸善	03-272-7211
〃	旭屋書店銀座店	03-573-4936
港区	書原新橋店	03-591-8738
〃	雄峰堂NS店	03-503-6586
〃	虎ノ門書房本店	03-502-3461
〃	虎ノ門書房田町店	03-454-2571
品川区	芳林堂大井町店	03-474-4946
〃	明屋書店五反田店	03-492-3881
渋谷区	紀伊國屋書店渋谷店	03-463-3241
〃	旭屋書店渋谷店	03-476-3971
〃	三省堂書店渋谷店	03-407-4545
〃	大盛堂書店	03-463-0511
〃	紀伊國屋書店笹塚店	03-485-0131
新宿区	紀伊國屋書店本店	03-354-0131
〃	三省堂書店新宿西口店	03-343-4871
〃	福家書店センタービル店	03-345-1246
〃	福家書店野村ビル店	03-342-0298
〃	新星堂NSビル店	03-344-2055
〃	西武新宿ブックセンター	03-208-0380
〃	芳林堂高田馬場店	03-208-0241
〃	未来堂	03-200-9185
豊島区	旭屋書店池袋店	03-986-0311
〃	芳林堂池袋店	03-984-1101
〃	リブロ池袋店	03-981-0111
〃	三省堂書店池袋店	03-987-0511
〃	新栄堂本店	03-984-2345
〃	新栄堂アルパ店	03-988-0181
台東区	明正堂中通り店	03-831-0191
墨田区	ブックストア・談	03-635-1841
葛飾区	文教堂青戸店	03-838-5938
江戸川区	文教堂西葛西店	03-689-3621
大田区	アクトブックスサンカマタ店	03-735-1551
〃	竜文堂大森駅ビル店	03-775-3851
中野区	明屋書店東京本社	03-387-8451
杉並区	ブックセンター荻窪	03-393-5571
〃	書原杉並店	03-313-4778
武蔵野市	紀伊國屋書店吉祥寺東急店	0422-21-5543
〃	弘栄堂吉祥寺店	0422-22-1031
〃	パルコブックセンター吉祥寺	0422-21-8122
調布市	真光書店	0424-87-2222
府中市	啓文堂	0423-66-3151
三鷹市	三省堂書店三鷹店	0422-48-4510
〃	東西書房	0422-46-0275
小金井市	文教堂小金井店	0423-86-0161
国分寺市	三成堂国分寺店	0423-25-3211

展示図書一覧

定価は本体価格です。

書名	定価
MS-DOSいたれりつくせり本	●1800円
プレイMS-DOS	●1900円
UNIX System V プログラマ・ガイド	●12000円
UNIX System V ユーザ・ガイド	●9800円
UNIXオペレーティングガイド	●3000円
OS/2 APIブックⅠ	●2709円
C言語の活用理解	●2000円
C言語の基礎知識	●2500円
C言語の応用50例	●2300円
上級・C言語の応用例50例	●2400円
Cプリプロセッサ・パワー	●2200円
Play the C 上・下	●各1500円
Turbo C入門	●2600円
C++プログラミング	●2600円
Quick Cプログラミング	●2602円
詳説C言語	●4369円
8086アセンブリ言語	●2800円
8086マクロプログラミング	●2600円
Final Ver.4.0ブック	●2400円
MIFES Ver.4.0ブック	●2400円
ビジネスソフトデータ活用ブック	●2800円
BASICによるプログラミングスタイルブック	●1800円
ソーティング・ノート	●1900円
J-3100パワーユーザーブック	●2400円
続・PC工作入門	●1800円
PC-286Lブック	●1700円
試験に出るX1	●2800円
RDBファラオ活用ガイド	●2903円
言図ガイド	●2301円
Rydeenガイド	●2427円
P1EXEガイド	●2524円
Lotus1-2-3ガイドⅡ	●2500円
MS-Chart Ver.3.1ガイド	●2900円
まいと〜くガイド	●2300円
新松ガイド	●2000円
一太郎Ver.3ガイド	●2500円
新一太郎ガイド	●2300円
桐Ver.2ガイド	●2500円
花子応用ガイド	●2500円
Lotus1-2-3ガイド	●2400円
P1ガイド	●2300円
Ninja2ガイド	●2300円
Multiplan Ver.3.1ガイド	●2400円
アセンブラCASL入門	●2000円
ハードウェア徹底マスター	●2500円
FORTRAN徹底マスター	●2800円
情報処理の基礎知識	●1600円
COBOL徹底マスター	●2900円
受験用語ハンドブック	●1800円
情報処理入門1・2	●各1204円
CASLで学ぶアセンブラ言語入門	●2204円
バイト&ワードの風にのって	●1800円
田原総一朗のパソコンウォーズ	●1400円
パソコンを襲う知的独占の戦い	●1600円
RPG幻想事典・日本編	●1800円
魔法王国シムルグント	●1800円

国立市	東西書店	0425-75-5061
小平市	文教堂小平店	0423-43-9229
東村山市	文教堂東村山店	0423-96-1115
立川市	オリオン書房ウイル店	0425-27-2311
八王子市	くまざわ書店本店	0426-25-1201
町田市	有隣堂町田店	0427-23-3018
〃	久美堂本店	0427-25-1330
〃	久美堂小田急店	0427-27-1111
〃	文教堂鶴川店	0427-35-4117
〃	文教堂小川店	0427-96-1781
多摩市	くまざわ書店桜ヶ丘店	0423-37-2531
福生市	文教堂福生店	0425-53-7708
〈甲信越・北陸〉		
甲府市	文教堂甲府店	0552-22-4600
長野市	平安堂長野店	0262-26-4545
〃	長谷川書店	0262-26-2122
上田市	平安堂上田店	0268-22-4545
松本市	ブックスロクサン	0263-35-5555
〃	改造社松本駅ビル店	0263-36-3777
飯田市	平安堂飯田店	0265-24-4545
岡谷市	笠原書店	0266-23-5070
諏訪郡	平安堂下諏訪店	0266-28-1111
新潟市	紀伊國屋書店新潟店	025-241-5281
〃	萬松堂	025-229-2221
〃	北光社	025-228-2321
長岡市	覚張書店	0258-32-1139
〃	ブックセンター長岡	0258-36-1360
〃	長岡技大長峰文化	0258-46-6437
上越市	パソトピア コスモス	0255-25-5867
山北町	BOOKメディア	0254-77-3850
富山市	瀬川書店	0764-24-4566
〃	清明堂	0764-24-4166
〃	BOOKSなかだ豊田店	0764-32-1353
〃	文苑堂本郷店	0764-22-0552
〃	文苑堂赤江店	0764-33-0321
高岡市	文苑堂	0766-21-0333
〃	文苑堂横田店	0766-21-0431
金沢市	うつのみや片町店	0762-21-6136
〃	書林香林坊本店	0762-20-5011
野々市町	王様の本本店	0762-46-5325
福井市	勝木書店	0776-24-0428
〃	品川書店新田塚店	0776-24-1112
〈東　海〉		
静岡市	静岡谷島屋呉服町本店	0542-54-1301
〃	江崎書店	0542-54-4481
〃	吉見書店	0542-52-0157
〃	戸田書店ＳＢＳ店	0542-81-5733
〃	戸田書店曲金店	0542-81-5899
沼津市	吉野屋	0559-23-5676
〃	マルサン書店宝塚店	0559-63-0350
富士市	戸田書店富士店	0545-51-5121
清水市	戸田書店本店	0543-65-2345
浜松市	浜松谷島屋連尺本店	0534-53-9121
名古屋市	三省堂書店名古屋店	052-562-0077
〃	星野書店近鉄ビル店	052-581-4796
〃	丸善名古屋支店	052-261-2251
〃	丸善ブックメイツセントラルパーク	052-971-1231
〃	日進堂上前津店	052-263-0550
名古屋市	三洋堂パソコンショップΣ	052-251-8334
〃	三洋堂いりなか本店	052-832-8202
〃	ちくさ正文館本店	052-741-1137
〃	白樺書房西店	052-774-7223
豊橋市	精文館	0532-54-2345
岡崎市	ブックス鎌倉	0564-54-1822
豊田市	三洋堂梅坪店	0565-35-2334
豊川市	三洋堂豊川店	05338-3-0334
刈谷市	三洋堂刈谷店	0566-24-1134
春日井市	三洋堂勝川店	0568-32-7806
岐阜市	自由書房	0582-65-4301
大垣市	大洞堂ブックス258	0584-81-2553
〃	大洞堂岐大バイパス店	0584-74-7766
一宮市	三洋堂一宮店	0586-77-5734
可児市	三洋堂可児店	0574-63-2334
多治見市	三洋堂多治見店	0572-24-0340
津市	別所書店11ビル店	0592-24-1014
四日市市	文化センター白揚	0593-51-0711
鈴鹿市	シェトワ白揚スズカ	0593-82-5221
〈近　畿〉		
京都市	駸々堂京宝店	075-223-1003
〃	アバンティ・ブックセンター	075-682-5031
〃	オーム社書店河原町店	075-221-0280
〃	ジュンク堂京都店	075-252-0101
〃	オーム社書店竹田店	075-644-2611
奈良市	駸々堂大丸店	0742-26-6241
大阪市	旭屋書店本店	06-313-1191
〃	紀伊國屋書店梅田店	06-372-5821
〃	オーム社書店大阪店	06-345-0641
〃	駸々堂京橋店	06-353-3209
〃	駸々堂心斎橋店	06-251-0881
〃	旭屋書店ナンバ店	06-644-2551
〃	ナンバブックセンター	06-644-5501
〃	ヒバリヤ書店ナンバ店	06-644-5407
〃	旭屋書店アベノ店	06-631-6051
〃	ユーゴー書店	06-623-2341
〃	河村書店	06-951-2968
枚方市	水嶋書房京阪デパート店	0720-51-3432
高槻市	コーベブックス西武高槻店	0726-83-1766
東大阪市	ヒバリヤ書店本社	06-722-1121
神戸市	ジュンク堂センター街店	078-392-1001
〃	ジュンク堂サンパル店	078-252-0777
〃	海文堂書店	078-331-6501
〃	日東館書林	078-391-8701
姫路市	新興書房	0792-85-3344
〃	誠心堂書店	0792-81-2055
和歌山市	宮井平安堂	0734-31-1331
〃	帯伊書店	0734-22-0441
〈中　国〉		
岡山市	紀伊國屋書店岡山店	0862-32-3411
〃	丸善岡山支店	0862-31-2261
津山市	津山ブックセンター	08682-6-4047
広島市	紀伊國屋書店広島店	082-225-3232
〃	丸善広島支店	082-247-2251
〃	金正堂	082-248-3715
〃	積善館	082-248-3151
尾道市	啓文社尾道店	0848-37-5151
福山市	啓文社福山店	0849-22-3111
福山市	ブックシティ啓文社	0849-25-0050
〃	啓文社コア	0849-41-0909
山口市	五十部誠文堂	0839-24-6630
〃	文栄堂	0839-22-5611
下関市	中野書店	0832-22-6181
宇部市	京屋書店	0836-31-2323
〃	末広書店	0836-31-0086
防府市	誠文堂国衙店	0835-25-1988
光市	三文字屋	0833-71-0251
鳥取市	富士書店	0857-23-7271
松江市	園山書店	0852-21-4167
〈四　国〉		
徳島市	小山助学館本店	0886-54-2135
〃	小山助学館東ロ店	0886-25-1380
〃	森住丸善	0886-23-3228
高松市	宮脇書店本店	0878-51-3733
丸亀市	宮脇書店丸亀店	0877-22-5533
松山市	紀伊國屋書店松山店	0899-32-0005
〃	明屋書店本店	0899-41-4141
〃	明屋書店大街道店	0899-41-4242
〃	丸三書店	0899-31-8501
新居浜市	明屋星原店	0897-44-4000
宇和島市	明屋宇和島店	0895-23-1118
高知市	金高堂	0888-22-0161
〈九州・沖縄〉		
福岡市	紀伊國屋書店福岡店	092-721-7755
〃	りーぶる天神	092-713-1001
〃	積文館新天町店	092-781-2991
〃	福岡金文堂本店	092-741-2106
〃	福岡金文堂朝日ビル店	092-431-1094
〃	福岡金文堂デイトス店	092-451-6175
〃	福岡金文堂アニマート原	092-844-0088
北九州市	ナガリ書店	093-521-1044
〃	金栄堂	093-531-3685
〃	旭屋書店北九州店	093-631-6421
〃	井筒屋ブックセンター	093-641-0131
〃	カルパーク平野	093-661-7988
〃	白石書店本城店	093-601-2200
久留米市	エマックスたがみ	0942-33-1841
飯塚市	BOOKリード	0948-25-7266
大分市	パルコブックセンター大分店	0975-35-0643
〃	本町晃星堂	0975-33-0231
別府市	明林堂	0977-23-2183
宮崎市	中央、田中書店	0985-24-3511
〃	寿屋宮崎店	0985-27-4111
佐賀市	金華堂北バイパス店	0952-32-1965
〃	積文館佐賀店	0952-24-4314
〃	積文館デイトス店	0952-23-7155
長崎市	メトロ書店	0958-21-5453
〃	好文堂	0958-23-7171
佐世保市	金明堂書店	0956-22-4214
熊本市	紀伊國屋書店熊本店	096-322-5531
〃	長崎書店	096-353-0555
人吉市	明屋人吉店	0966-22-5486
鹿児島市	春苑堂ブックプラザ	0992-25-3200
〃	ブックスみすみ	0992-57-1011
那覇市	球陽堂書房ビル店	0988-63-3752
〃	文教図書	0988-62-1201

大学生のための
パソコン業界研究セミナー

SOFTBANK NEWS

1990年(平成2年)7月1日

ソフトバンク株式会社
〒108 東京都港区高輪2-19-13 NS高輪ビル
TEL:03-5488-1115

LAN業界大手企業との合弁事業大規模展開

ソフトバンク株式会社

ソフトバンク(株)は米国最大のLANソフト専門会社、ノベル社と提携。ノベル株式会社を発足させた。米国より遅れている日本のLAN市場が大きく変貌する可能性がある。

四月十二日(木)、ノベル株式会社設立発表会が新高輪プリンスホテルに於いて開催された。当日、ノベル(株)設立の記者会見、「ノベルネットウェアの現状と展望」「日本でのネットウェア・コンソーシアムについて」と題し、設立記念パーティーも盛りだくさんで行われた。



主催 株式会社 日本ソフトバン

ネットウェアの日本語化と販売や、ノベルの代理店の窓口的役割を果す等。すでに日本に存在しているネットウェアはネットワークで結んだパソコンやワークステーション間での情報のやり取りを管理するソフトであり、米国では六十%以上のシェアを持つ。ネットウェアの日本語化により、今後のLAN普及が期待される。

ネットウェアはネットワークで結んだパソコンやワークステーション(WS)間で情報をやりとりするための基本ソフト(NOS)。日本では米国に比べLANの普及が遅れていた。しかし、ここ一、二年で導入する企業が増え、LAN市場が拡大し始めていることから、今後注目できる市場といえる。

PCWEEK 日本語版創刊

ソフトバンク出版事業部では、五月十六日、パソコン情報誌「PC WEEK」日本語版を創刊した。「PC WEEK」は、アメリカにおけるコンピュータ関連の出版では最大規模のジフ・デイビス社との全面提携によるものである。タブロイド版で約四十ページ。四色使用のグラフィックチャート・グラフ・イラストなどを掲載、オールカラーでクォリティーを追求したものとなっている。また創刊と同時にBBSシステムを立ち上げ、読者に対してIDを提供し、読者の意見や主張を紙面に反映する等、双方コミュニケーションを目指している。週刊で毎週五万部を予定している。



九十年度新卒入社員集合研修開催

フレッシュな新人が夢と希望を抱いて陽気なひとときを過ごす!

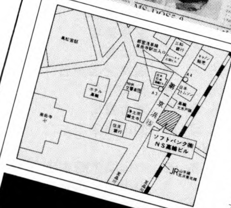
四月二日より八日にかけ、山梨県山中湖ホテルにてフレッシュマンたちの集合研修が行われた。今年はソフトバンク(株)史上最多の男四十六名、女三十六名、計八十二名の新卒採用となった。個性も千差万別で、趣味はドライブからカラオケ、料理、ゴルフ、中国武術まで多種多様な一面がうかがえる。彼らの意識調査をしてみると、「結婚したい年齢」は男女共に二十五才〜二十七才が一番多く、また「海外旅行で行きたい所」はヨーロッパ、アメリカ、オセアニアの順であった。



フレッシュな息吹を会社に運んでくれた彼ら。急成長するソフトバンク(株)で更に磨かれ、近い将来、世界をまたにかけるジャパニーズ・ビジネスマンが彼らの中から育つのであろう。会社の期待も大きい。

日本ソフトバンク社名を「ソフトバンク」と改称

(株)日本ソフトバンクは九十二年の株式上場に備え、社内体制の強化に着手している。国際的事業展開に向け、社名も「ソフトバンク株式会社」と改称した。六月末には東京・高輪に本社を移転。都内八ヵ所の建物に分散していた各部門(ソフト流通、出版、通信、管理)を統合した。新住所は、東京都港区高輪二―十九―十三 NS高輪ビル。



セミナー開催日決定!

熱烈待望!電界講習!
ーソフトバンク株式会社セミナー
ーただ今…
いる

ソフトバンク株式会社 パソコン業界研究セミナー日程

会場		連絡先	
札幌	7月23日	TEL 011 (222) 6026	札幌営業所
仙台	7月20日	TEL 022 (263) 0907	仙台営業所
東京	7月16、24日	TEL 03 (5488) 1115	本社人事部
名古屋	7月26日	TEL 052 (261) 7215	名古屋営業所
大阪	7月9日	TEL 06 (264) 1471	西日本営業部
広島	7月11日	TEL 082 (223) 1314	広島営業所



ソフトバンク

〒108 東京都港区高輪2-19-13
NS高輪ビル
TEL 03 (5488) 1115

HOST

X68000専用
多回線 ホストソフト

PRO-68K

ついに登場!

3回線/9回線

きみも、今日から局長さん

HOST 9 PRO-68K 概要

対応回線数	1〜9回線
使用モデム	ATモデム MNP(RTS/CTS)可
通信速度	最大9600bqs
会員数	*最大9999人
掲示板数	*最大40個
機能	電子掲示板・電子手数・電子会議(チャット)・会員情報

これらは、コンフィグファイルで設定できます。

注1:*印について拡張を希望する場合は、プログラムの書き換えが必要になりますので、別料金にて対応致します。当社までご相談ください。
2:2回線以上で運用される場合は、CZ-6BF1(シャープ純正)が必要になります。
3:このホストはテキスト形式の転送方法を採用しております。

■特長

●各種設定のコンフィグファイル化。●RS-232C回線とは別にキーボードからのアクセス、ダウンロード、アップロードが可能。●モニタで、各チャンネルのユーザーの打ち込んだコマンドや通信状態を確認。●各掲示板別にSIG、ボードパスの設定。●メンテナンス作業のオンライン実行。(ボードインデックス,メールインデックス)●オンラインサインアップ等、ゲストへの設定が可能。●通信サービスTri-P対応。●行編集(オンライン簡易エディタ)機能。●その他・シスオペレベルで会員情報の変更が可能。タイムアウトによる回線切断。PDS専用掲示板の採用。(1書込中で、ドキュメントとテキストプログラムの分離)。●接続MNPタイプの識別。●ログイン、ログアウト時間の記録。●非アクセス時のモニタ画面消去可能。

HOST 3 PRO-68K

機能は統べて、「HOST 9 PRO-68K」と同じですが、対応回線数が、1〜3回線に制限されて、低価格でユーザーに供給します。

バージョンアップ(Ver1.10) サービス実施中

現在発売されています製品は、Ver1.10に変更になっています。お使いの製品がVer1.00のユーザーの方のために、バージョンアップサービスを実施しておりますので、お早目に、ユーザー登録葉書をお送り下さい。
Ver1.10へ無料交換を実施しております。

好評発売中

HOST 9 PRO-68K ¥59,800円
HOST 3 PRO-68K ¥39,800円

SPS-NET
TSUKUMO-NET モデル運用中!!

今、X68000の通信が変わる!!!

た〜みのる 2

ユーザー重視の機能を搭載して
好評発売中
17,800円

24/31KHz ディスプレイ対応

「た〜みのる」が
装いも新たに
「た〜みのる2」として登場!
「た〜みのる」が
通信入門版なら
「た〜みのる2」は
マニアタイプの通信ソフトです!!!

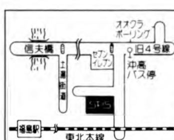
X68000専用
パソコン通信ソフト

「た〜みのる2」はX68000用に製作された通信ソフトです。X68000の機能を充分に引き出して、ユーザーの方々が簡単に操作できるよう工夫・製作されています。

プログラマ募集!!

SPSでゲームを作ってみませんか?

アセンブラでプログラムの組める優秀な人材を若干名募集しています。就職希望の方は62円切手同封の上、「就職案内係 大和」までお手紙ください。折り返し就職のご案内をお送り致します。
尚、デザイナー、音楽プログラム等の専門職は募集しておりません。



(株)マイコンハウス
SPS
〒960 福島市太平寺字町ノ内5-3 ☎(0245)45-5777
FAX(0245)45-1804(GII,GIII)

■表示価格に消費税は含まれておりません。

当社の製品は全国の有名デパート、パソコンショップでお求めになれます。尚、お求めになれない場合、郵便局にてお申し込みください。●口座番号 郡山5-12298 ●加入者名(株)エス・ピー・エス ●金額 代金に3%の消費税を加算した額●通信欄(裏面)ご希望ゲームソフト名、数量、代金合計、年齢、氏名、機種名、テープかディスクの種類。(一週間以上かかりますので、お急ぎの方は現金書留をご利用ください。その場合、おつりのいらないようにお願いします。

X68000 HOST PRO-68K 使用 **SPS-NET** TEL (0245)46-1167(代)

Tri-P 好評運営中(4回線)/一般回線(9回線) MNPクラス7

24時間運営(N81XN)
ゲストID(GUEST)

※GUESTアクセスは無料ですのでぜひ、一度試してください。

入会方法

登録料¥3,000(税別)
会費無料

下記の用紙に直接記入するか又は、コピーして記入し、72円切手同封の上、「SPS-NET係」までお送り下さい。届き次第、仮登録を行いID発行後SPS-NET専用の郵便振込み用紙ならびに運用の手引きをお送りいたします。それに従い、3ヶ月以内に登録料3,000円(税別)を御入金下さい。
入金確認後正式会員として再登録します。

例◎パスワード=SPS-NET (8文字まで大小文字の識別あり)
( )

◎職業=株式会社エス・ピー・エス(16文字まで)
( )

◎本名=大和大五郎(8文字まで)
( )

◎住所=福島市太平寺字町ノ内5-3(24文字まで)
(〒)

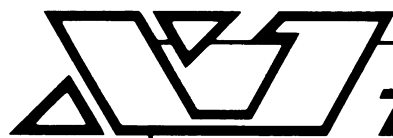
◎ペンネーム=大ちゃん(4文字まで)
( )

◎自己紹介=SPS-NETをよろしく (24文字まで)
( )

◎年齢=30(現在の年齢)
( )

◎システム構成=X68000ACE-HD MD2400B (18文字まで)
( )

◎電話=0245-45-5777(市外局番から)
( )



専用 turbo OK-システム漢字

「個人簿記会計 財計くん」2HD版
定価 49,800円（税別）

出力帳票：勘定科目一覧表・摘要一覧表・期首貸借対照表・期末試算表・貸借対照表・損益計算書・仕訳帳・各科目別元帳・合計残高試算表

処理金額	9桁10億円/年間
月間仕訳処理数	900件以内
仕訳入力は一度	振替伝票方式採用
使用勘定科目数	75個(年度変更可)
摘要小書き入力	A・Bの2つ Aはコード入力 Bは自由入力
オート・ソート	仕訳訂正で 日付自動処理
ラクラク金額入力	カンマ付き、無 どちらもOK！

消費税の会計処理　注目の消費税の会計処理は、4つの対応が考えられますが、ユーザー別に勘定科目の設定をする事により処理できます。

「消費税検証」を別冊にて同梱してあります。ご活用下さい。

〈各種税法は変化しても、複式簿記の原理は不変です。勘定科目の設定によって処理できるのが、財計くんなのです。〉

プリンター用紙

縦11インチの白紙又は罫線入りを使用願います。

2D版との能力アップの内容

1．ディスクの入れ替えなしで、システムユーザー辞書使用可。
2．科目&摘要の入力時にHTLPキー機能を追加。

「個人簿記会計 財計くん」2D版
定価 39,800円（税別）

2HD版との相違は、先の能力アップの内容の通りです。

各資料のご請求は

資料は、一部あたり200円分の切手を同封願います。各デモ・サンプル版は実費2400円を申し受けます。

弊社へ直接お申込みの方は上記分を差し引いてご本体を購入できます。

資料は毎月曜日に、デモ版は逐次発送しています。

「財計くん 売掛管理台帳」2HD版
定価 39,000円（税別）

出力帳票：納品書・請求書・アイウエオ順顧客一覧表・取扱商品一覧表・売上日計表・売掛残高一覧表・DMシール（条件検索可）

処理金額	9桁　10億円/年間
1顧客処理件数	60件/月間　繰越可
処理顧客数	1DataDisk　1200名
取扱商品数	1DataDisk　250品目
消費税自在処理	登録済使用と未登録 使用どちらも可
登録済顧客変更	台帳変更Bで自在
帳票3段階選択	顧客別&〆切&全部
商品単価無登録	250品目が無限に
ラクラク金額入力	カンマ付き、無 どちらもOK！

プリンター対応表

ご使用になる機種により4つのシリーズ品番がございます。ご購入の際にはご確認願います。

No701：CZ－8PK3・CZ－8PK4・CZ－8PK5・C－8PK6・CZ－8PK7・CZ－8PK8・CZ－PK9・EPSON－VPシリーズ＝X1ROM要
No702：CZ－8PK2・CZ－80PK
No703：CZ－8PD2・CZ－8PD2・CZ－800P・EPSON－SPシリーズ＝X1ROM要
No704：X1に接続可能なもので、縦11インチの白紙又は罫線入りのものみを利用する事になります。

＊　伝票専用用紙として、ヒサゴ(株)GB－342を使用します。伝票以外は縦11インチの連続用紙(白紙or罫線入り)を使います。 なお、No.704のみは、伝票用紙はユーザーが作成して使用する事になります。

2D版との能力アップの内容

1．ディスクの入れ替えなしで、システム・ユーザー辞書使用可。
2．商品名の入力時にHELPキー機能が追加。

「財計くん 売掛管理台帳」2D版
定価 29,000円（税別）

2HD版との相違は、先の能力アップの内容の通りと、処理顧客数が600名となり、取扱商品数が150品目となります。（2HD同様No701～No.704品番がございます。ご購入の際はご確認下さい。）

「DATA・CARD 1200」 2HD版
定価 42,000円（税別）

カード型データーベースとしての機能とグラフ作成ツールのグラフデーター・ファイル機能を持っています。検索は、1,124枚のデーターカードから3重条件を処理します。

項目設定は自由設定で12個までを処理し、データー部は新規に設けました「データー変換Uty」で、作成済みのデーターでもデーター量に応じて変更可能になりました。

DMシール発行・葉書宛名印刷を条件検索で処理します。

カードNoによる、データーの抜粋・ステップ印刷（同カードを最大12枚まで）を処理します。

グラフ・ツールとしては、7種・22タイプのグラフを作成する事ができ、最大12項目12データーを縦棒グラフ・横棒グラフ・帯グラフ・円グラフ・折線グラフに処理します。縦棒グラフ・横棒グラフは3D仕様でも処理します。

プリンター用紙

縦11インチの白紙又は罫線入りを使用願います。

2D版との能力アップの内容

1．ディスクの入れ替えなしで、システム・ユーザー辞書使用可。
2．グラフDataDisk内に格納できるファイル数が3倍になりました。

「DATA・CARD 1200」 2D版
定価 32,000円（税別）

2HD版との相違は、先の能力アップの内容の通りです。

ご購入は

お近くのパソコン・ショップでお求め下さい。お急ぎの方は直接現金書留でお申し込み下さい。

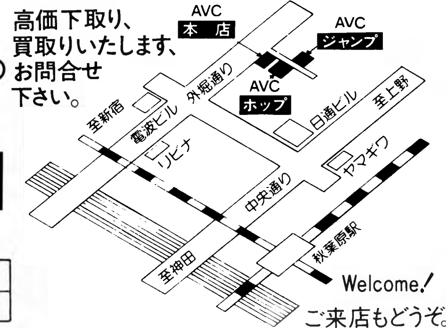
（売掛管理台帳のNo704のみユーザーのご希望により、プログラム解放型2D￥58,000円（税別）もあります。直接弊社にお申し込みください。）

〒885　宮崎県都城市都島町430－2
OK－ハウス
TEL 0986－25－0303－FAX 0986－25－9553





〒101 東京都千代田区外神田3-2-3 ☎03-253-7611(代)



今すぐ もよりの電話から	仙 台 022-264-3704	名古屋 052-452-3271	広 島 082-295-6873
札 幌 011-611-5104	新 潟 0252-75-4175	大 阪 06-311-3931	福 岡 092-481-2494

X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

X68000 待望の新しい仲間登場!!

X68000 PERSONAL WORKSTATION EXPERTII・EXPERTII HD



CZ-603C 標準価格¥338,000
CZ-613C 標準価格¥448,000
AVC特価

X68000 PERSONAL WORKSTATION PROII・PROII HD



PROII・PROII HD
拡張I/Oポートを4スロットを搭載し、汎用性と低価格が魅力。
もちろん、SX-WINDOW搭載。

CZ-653C 標準価格¥285,000
CZ-663C 標準価格¥395,000
AVC特価

X68000

在庫稀少価格はお電話で!

CZ-602C
CZ-612C
CZ-652C
CZ-662C

お勧めディスプレイコーナー　組合せは自由、価格はお気軽にご相談下さい。

CZ-604D 標準価格¥94,800 **AVC特価**
- 0.31mmドットピッチ
- 2モードオートスキャン
- ステレオスピーカ搭載
- チルト台同梱

CZ-613D 標準価格¥135,000 **AVC特価**
- ドットピッチ0.31mm
- TVチューナー搭載
- ステレオスピーカー搭載
- チルト台同梱

CZ-603D 標準価格¥84,800 **AVC特価**
- 0.31mmドットピッチ
- TVチューナ無し
- 3モードオートスキャン
- チルト台同梱

CU-21HD 標準価格¥148,000 **AVC特価**
- 0.52mmドットピッチ
- 21型ディスプレイ
- 3モードオートスキャン
- ステレオスピーカ搭載

CZ-605D 標準価格¥115,000 **AVC特価**
- ドットピッチ0.39mm
- TVチューナー搭載
- ステレオスピーカー搭載
- チルト台同梱

CZ-602D 標準価格¥99,800 **AVC特価**
- ドットピッチ0.39mm
- TVチューナー搭載
- チルト台同梱

型 番	品 名	標準価格	販売価格
CZ-6TU	システムチューナー	¥ 33,100	AVCフタバ特価
BF-68PRO	CRTフィルター	¥ 19,800	AVCフタバ特価
CZ-8NS1	カラースキャナー	¥188,000	AVCフタバ特価
CZ-6BN1	スキャナー用パラレルボード	¥ 29,800	AVCフタバ特価
CZ-6VT1	カラーイメージユニット	¥ 69,800	AVCフタバ特価
CZ-8BV2	カラーイメージボード	¥ 39,800	AVCフタバ特価
CZ-8BR1	立体映像セット	¥ 29,800	AVCフタバ特価
CZ-8DT2	パーソナルテロッパ	¥ 44,800	AVCフタバ特価
CZ-8BS1	FM音源ボード	¥ 23,800	AVCフタバ特価
CZ-8NJ1	ジョイカード	¥ 1,700	AVCフタバ特価
CZ-8NM2A	マウス	¥ 6,800	AVCフタバ特価
CZ-8NM3	マウス・トラックボール	¥ 9,800	AVCフタバ特価
CZ-6SD1	システムラック	¥ 44,800	AVCフタバ特価
AN-S100	アンプ内蔵スピーカー	¥ 36,600	AVCフタバ特価
CZ-6EB1	拡張I/Oボックス	¥ 88,000	AVCフタバ特価
CZ-8PG1	24ピンカラープリンター(80桁)	¥130,000	AVCフタバ特価
CZ-8PK10	24ピンプリンター(136桁)	¥ 97,800	AVCフタバ特価
IO-735X	カラージェットプリンター	¥248,000	AVCフタバ特価
CZ-6BE1A	1M増設RAMボード	¥ 38,000	AVCフタバ特価
CZ-6BE2	2M増設RAMボード	¥ 79,800	AVCフタバ特価
CZ-6BE4	4M増設RAMボード	¥138,000	AVCフタバ特価
CZ-6BP1	数値演算プロセッサー	¥ 79,800	AVCフタバ特価
CZ-6BC1	FAXボード	¥ 79,800	AVCフタバ特価
CZ-6BM1	MIDIボード	¥ 26,800	AVCフタバ特価
CZ-6BU1	I/Oボード	¥ 39,800	AVCフタバ特価
CZ-6BL1	LANボード	¥268,000	AVCフタバ特価
CZ-243BS	サイバーノート	¥ 19,800	AVCフタバ特価
CZ-240BS	スティショナリー	¥ 14,800	AVCフタバ特価
CZ-223CS	通信ソフト	¥ 19,800	AVCフタバ特価
ゲームソフト		20% OFF	
CZ-8TM2	モデムユニット	¥ 49,800	AVCフタバ特価
CZ-252MS	Musicstudio	¥ 28,800	AVCフタバ特価
CZ-247MS	MUSIC(MID)	¥ 28,800	AVCフタバ特価
CZ-221HS	NEW Print Shop	¥ 19,800	AVCフタバ特価
CZ-228BS	TOP給与計算エキスパート	¥200,000	AVCフタバ特価
CZ-227BS	TOP財務会計	¥200,000	AVCフタバ特価
CZ-220BS	DATA	¥ 58,000	AVCフタバ特価
CZ-212BS	BUSINESS	¥ 68,000	AVCフタバ特価
CZ-219SS	OS-9	¥ 29,800	AVCフタバ特価
CZ-211LS	Ccompiler	¥ 39,800	AVCフタバ特価
CZ-234LS	AI-68K	¥188,000	AVCフタバ特価
CZ-620H	20MBハードディスク	¥178,000	AVCフタバ特価
CZ-64H	40MBハードディスク	¥120,000	AVCフタバ特価
LHD-34V	40MBハードディスク(ロジテック)	¥153,000	¥117,000
LHD-32V	20MBハードディスク(ロジテック)	¥128,000	¥ 98,000

CZ-8NJ2



アナログジョイステック
標準価格 ¥23,800

AVC特価¥???

CZ-8PG2



24ピンカラー。
漢字ドットインパクトプリンタ

CZ-8PG2……¥160,000

AVC特価¥???

CZ-8PC4



48ドット熱転写プリンター。精密な文字、ハードコピーも可能。

CZ-8PC4……¥ 99,800

AVC特価 ¥64,800

CZ-8PC3



24ドット熱転写カラープリンター

標準価格……¥65,800

AVC特価 ¥39,800

●但し消費税(3%)は別途請求させていただきます。●分割回数は3回～48回まで自由に選べます。

●セットの組合せは自由!広告に出ていない他の機種はお問合せ下さい。

●頭金なし(手軽な電話クレジット)●製品先取り(お支払いは約1～2ヶ月後から)●低金利クレジット(1回の支払いは2,700円以上で3～48回。ボーナス併用も可)●カレッジクレジット(保証人なし。但し満20歳以上の学生の方)●18歳未満の方(ご両親が代理購入者としてお申し込み下さい)
●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますので御了承下さい)
●完全保証(すべてメーカー保証書付。アフターケア万全)●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

AM10時からPM7時まで受付 日曜・祝日も営業

信用と実績を誇る
BASIC HOUSE

BASIC HOUSE 大田原店

一周年 大謝恩セール！

〈主な取扱いメーカー〉

Apple Computer
FUJITSU
NEC
EPSON
SHARP
TOSHIBA

全品大特価!!

7/20(金)〜7/23(月)

BASIC HOUSE 宇都宮

店内改装のため

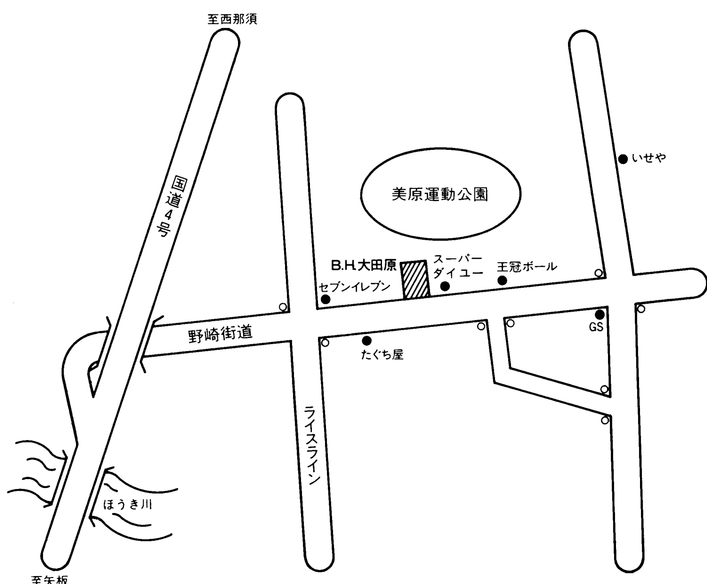
★閉店売りつくしセール

7/20(金)〜7/23(月)

★新装開店セール

8/3(金)〜8/6(月)

※通信販売もOKです!!



全国どこでも発送可　長期クレジットＯＫ　送料全国均一￥1,000　宅配便にて即日配送

株式会社**計測技研**

本社営業部／マイコンショップ／通販部	宇都宮市竹林町503—1	TEL0286-22-9811	FAX0286-25-3970
大田原営業所／マイコンショップ	大田原市美原1—13—4	TEL0287-23-5352	FAX0286-23-5364

マイコンショップ **BASIC HOUSE** お申し込み・お問い合せは ☎0286-22-9811(代)

2枚のボードが1枚になった

KGB-X68PRK



※写真はKGB-X68PRK-14です

広大なメモリ空間を実現する最大4Mバイトの

高速増設メモリ

高速演算を約束してくれる

数値演算プロセッサ

- メモリアクセスノーウェイトによる高速アクセス
- CZ-6BE2/CZ-6BE4/CZ-6BP1との混在が可能です
- 複数枚のKGB-X68PRKの実装が可能です
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能です
- ジャンパの変更により数値演算プロセッサの1枚目2枚目/未使用の選択が可能です
- 1M/2M/3Mメモリモデルは購入後にメモリをボード上に追加可能です
- 数値演算プロセッサにはデバイスドライバ(FLOAT3X)が付属します

※CZ-602C/CZ-612C以外の機種ではCZ-6BE1/CZ-6BE1Aを実装している必要があります
※メモリアクセスノーウェイトのため拡張I/O BOXでは動作しません

製品価格一覧

製品	価格	製品	価格
KGB-X68PRK-01 (1Mメモリ/数値演算プロセッサ無し)	¥58,000	KGB-X68PRK-11 (1Mメモリ/数値演算プロセッサ付き)	¥96,000
KGB-X68PRK-02 (2Mメモリ/数値演算プロセッサ無し)	¥74,000	KGB-X68PRK-12 (2Mメモリ/数値演算プロセッサ付き)	¥112,000
KGB-X68PRK-03 (3Mメモリ/数値演算プロセッサ無し)	¥98,000	KGB-X68PRK-13 (3Mメモリ/数値演算プロセッサ付き)	¥136,000
KGB-X68PRK-04 (4Mメモリ/数値演算プロセッサ無し)	¥122,000	KGB-X68PRK-14 (4Mメモリ/数値演算プロセッサ付き)	¥160,000

購入後の増設費用

メモリ	
1Mバイト	¥24,000
2Mバイト	¥51,000
3Mバイト	¥76,000
数値演算プロセッサ MC68881RC16	¥38,000

充実のBASIC HOUSEハードウェア&ソフトウェア

製品	価格	製品	価格
高速12BIT,16CH A/Dコンバータボード(KGB-AD12) X1	¥118,000	高速12BIT,4CH D/Aコンバータボード(KGB-DA4) X1	¥98,000
フォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	¥42,000	汎用ローコストA/D&PIOボード(KGB-X1S) X1	¥19,800
ハードディスクインターフェースボード(KGB-HDIF) X1	¥16,000	高速12BIT,16CH A/Dコンバータ(KGB-X68ADC) X68000	¥128,000
アイソレーション16BITデジタル入出力ボード(KGB-X68PIO) X68000	¥68,000	64180CPUボードMach180(KGB-CPXB) X68000	¥98,000
ハンディプリンタ&インターフェース(HANDYPRINTjack) X68000	¥24,800	ローコストMIDIインターフェース(MELODY BOX) X68000	¥16,800

製品	価格	製品	価格	製品	価格
BASIC拡張関数パッケージ(B6-6301)	¥9,800	C言語ライブラリ(B6-6305)	¥6,800	BASIC拡張関数パッケージC言語ライブラリ付(B6-6306)	¥14,800
ディスクキャッシャー(B6-6304)	¥6,800	Toys & Tools (B6-6307)	¥6,800	アイコンエディタ(B6-6303) ¥4,800 / CP/M68Kエミュレータ(B6-6302)	¥19,800

PRKニューバリエーション販売開始! PRK10コプロセッサ付/メモリー無し 定価¥72,000

BASICHOUSE BBS TECOSYS NET TEL 0286-27-1829 /1200/2400ボーMNPクラス5/8ビット/パリティ無し/X制御無し ゲストIDなし(オンラインサインアップ)

全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配送

株式会社計測技研

本社営業部/マイコンショップ/通販部 宇都宮市竹林町503—1 TEL0286-22-9811 FAX0286-25-3970

大田原営業所/マイコンショップ 大田原市美原1—13—4 TEL0287-23-5352 FAX0286-23-5364

マイコンショップ BASIC HOUSE お申し込み・お問い合せは ☎0286-22-9811(代)

アイ・ツー EXE CLUB

新規ユーザー・EXE会員 大集合

★X68000ユーザーニーズに対応したハード・ソフト・ウエア・周辺機器は全て展示しています。

★新製品情報・ユーザー同士の情報交換ができる、メンバー様の憩いのスペースです。

★大特価セール期間中X68000・ディスプレイ・プリンター御購入の方は全国どこでも送料無料!!

★遠くでなかなかお越し頂けない方にも通販専用TELで専門スタッフ(X68 PRO STAFF)が親切丁寧にお答えします。

★X68000お買い上げの方、アイツーよりBigプレゼント。

X68000 オリジナルステッカー
X68000 フロッピータイトルシール
X68000 オリジナルテレフォンカード
X68000 バッグ
お好きなもの2点もれなくついてくる!!

★現在シャープX68000 EXE会員の方、おトモダチをご紹介下さい。ご購入成立時点でアイ・ツーとシャープよりステキなプレゼント進呈中!!

★アイ・ツーメンバーズ優待制度実施
アイ・ツーでX68000・及びソフトウエア周辺機器をお買上げ頂きましたユーザー様にはオリジナルメンバーズカードを送付致します。メンバーズの方には楽しいパソコンライフをおくれますように最善のフォローをアイ・ツーより提供します。

新製品入替機 大放出!!

展示品処分祭 早い者勝ち

EXPERT SET
CZ-612CBK
CZ-603DBK

40% off

定価+3%=¥567,324

↓

あなたのお支払はタッタの ¥338,000 ポ・ッ・キ・リ

シャープ販売コンテスト・パソコン部門最高峰賞
シャープ販売第一位受賞感謝セール!
■期間/7月18日〜8月17日



アイ・ツーinシャープグランドフェア'90 OSAKAスタジアムに多数のご来場頂きまして、誠にありがとうございました。アイ・ツーサンクスフェアPart2も只今企画中ですので、乞うご期待!!

X68000プロショップ(専門店)ならではの企画です!!

SX-WINDOW ってなぁ〜に?

X68000ユーザーみんな集まれ!
SX-WINDOWの勉強会?を開催しまーす。

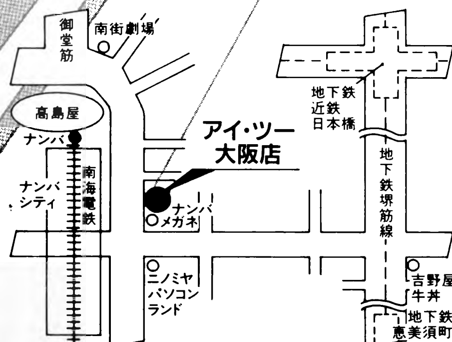
参加ご希望の方は、62円切手同封のうえ、お名前・ご住所・TEL・生年月日・お持ちのX68000の型番を書いて、アイ・ツーEXE CLUBあてで、おくって下さい。日時、場所etc...ご連絡します!!

場所はとりあえず大阪です!

X68000ユーザーとっておきのグッズ!!
X68000ユーザーのステータスシンボル。
新グッズもグループインしてますます充実。
キミのパソコンライフが一層楽しくなるコレクションだ!
X68000オリジナルグッズをまだ持っていないキミ
アイ・ツー からお届けしちゃいマス!



チャンスです 逃がす手はない



■営業時間 AM11:00〜PM8:00

通販専用TEL.
06-633-9800

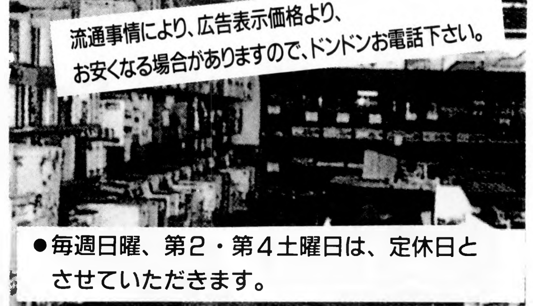
年中無休

Information & Interface
株式会社アイツー

大阪店/〒542 大阪市中央区難波千日前15-18

パソコン専門 O.A.ランド

アフターサービス万全
のサポート体制
優良パソコン販売店



●毎週日曜、第2・第4土曜日は、定休日とさせていただきます。

●お近くの方は、お立寄り下さい。専門係員がアドバイスいたします。
●ビジネスソフト、ゲームソフトのことならおまかせ下さい!!

セール期間
◀'90 7・15 ⟝ 8・15

サマーセール!! ドカ〜ンとプレゼント
OAランド恒例・大お買徳セール実施中

SHARP X68000シリーズセット (お楽しみゲームパック付)

●次代のインテリジェンス=SX-WINDOW搭載!!

X68000 EXPERTII
- CZ-603C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計¥453,000
OAランド大特価

クレジット	12回	¥30,200	24回	¥15,900



NEW

X68000 EXPERTII-HD
- CZ-613C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計¥563,000
OAランド大特価

クレジット	12回	¥37,400	24回	¥19,700

●SX-WINDOW搭載!!

X68000 PROII
- CZ-653C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

NEW

定価合計¥400,000
OAランド大特価

クレジット	12回	¥26,600	24回	¥14,000



X68000 PROII-HD
- CZ-663C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計¥510,000
OAランド大特価

クレジット	12回	¥33,800	24回	¥17,800

●SX-WINDOW搭載!!

X68000 SUPER-HD
- SX-WINDOW搭載
- SCSIインターフェース装備
- 80MBハードディスク搭載
- 3MB大容量メモリ装備
- 高解像度グラフィック

クレジット	12回	¥40,600	24回	¥21,400

NEW



X68000 SUPER-HD
- CZ-623C-TN(チタン)
- CZ-613D-TN(チタン)
- MD-2HD 20枚

定価合計¥633,000
OAランド大特価

X-1ターボZIIIセット

安すぎてゴメンなさい!

Ⓐセット
- CZ-888CBK…定価¥169,800
- CZ-880DBK‥ 定価¥109,800
- CZ-6ST1-B…定価¥ 5,800 (チルトスタンド)
- MD-2HD 20枚サービス

合計定価¥275,400
特価中TEL下さい



Ⓑセット
- CZ-888CBK…定価¥169,800
- CZ-830DBK…定価¥ 98,000
- CZ-6ST-1B…定価¥ 5,800 (チルトスタンド)
- MD-2HD 20枚サービス

合計定価格¥273,600
特価中TEL下さい

今月の特価品(限定)お早目に!!

★CZ-652C(BK)+CZ-602D(BK)
4セット限り……大特価**¥258,000**

★ CZ-612C(BK)
3セット限り…大特価**¥298,000**

- SHARP WD-A300(ワープロ) 定価¥165,000……特価¥110,000
- SHARP WD-A330(ワープロ) 定価¥185,000……特価¥125,000
- SHARP WD-HL30(ワープロ) 定価¥198,000……特価¥134,000
- SHARP PW-910(ワープロ)……特価¥ 85,000
- NEC PC-KD853(アナログCRT)……特価¥ 50,000
- 三菱XC-1498C(アナログCRT)……特価¥ 54,800
- SHARP CU-14FD(アナログCRT)……特価¥ 46,000
- SHARP PA-8500(電子手帳)……特価¥ 16,000

周辺機器コーナー

プリンターセットコーナー

- CZ-6PV1(カラービデオプリンター) 定価¥198,000……▶特価¥152,000
- CZ-8PC3(24ドット熱転写カラープリンター) 定価¥ 65,800……▶特価¥ 53,000
- CZ-8PK10(24ピン漢字ドットプリンター・136桁) 定価¥ 97,800……▶特価!/TEL下さい!
- CZ-8PG1(24ピンカラー漢字ドットプリンター・80桁) 定価¥130,000……▶特価!/TEL下さい!
- CZ-8PG2(24ピンカラー漢字ドットプリンター・136桁) 定価¥160,000……▶特価!/TEL下さい!
- IO-735X(カラーイメージェットプリンター) 定価¥248,000……▶特価!/TEL下さい!

OAランド特選品!!



■CZ-8PC4(定価¥99,800)
●48ドット熱転写カラー漢字プリンター 特価~~¥64,800~~

X68000用ソフトウェアー・コーナー

① CZ-212BS(BUSINESS)	定価¥ 68,000	▶特価¥ 53,000
② CZ-220BS(DATA)	定価¥ 58,000	▶特価¥ 45,000
③ CZ-215MS(Sampling)	定価¥ 17,800	▶特価¥ 13,800
④ CZ-221HS(NEW Print Shop)	定価¥ 10,800	▶特価¥ 15,500
⑤ CZ-227BS(TOP財務会計)	定価¥200,000	▶特価¥158,000
⑥ CZ-226BS(CARD)	定価¥229,800	▶特価¥ 23,000
⑦ CZ-223CS(Communication)	定価¥ 19,800	▶特価¥115,500
⑧ CZ-213MS(MUSIC)	定価¥ 18,800	▶特価¥ 14,800
⑨ CZ-211LS(C compiler)	定価¥ 39,800	▶特価¥ 31,000
⑩ C-TRACE(キャスト)	定価¥ 68,000	▶特価¥ 52,000
⑪ EW(イースト)	定価¥ 38,000	▶特価¥ 29,000

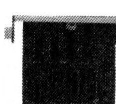
X68000用周辺機器コーナー

●CZ-6PU1A	定価¥ 38.000	▶特価¥ 30,000
●CZ-6BM1	定価¥ 26.800	▶特価¥ 21,000
●CZ-6BE1	定価¥ 88.000	▶特価¥ 69,800
●CZ-6VT1	定価¥ 69.800	▶TEL下さい
●CZ-8NS1	定価¥188.000	▶特価¥149,000
●CZ-6BC1	定価¥ 79.800	▶特価¥ 63,000

- 最新ゲームソフト その他各種ソフト 20%〜25%OFF!!
- 周辺機器・プリンター 割引販売中!! TEL下さい!

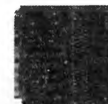
■I・O DATA 増設RAMボード

●1MB増設RAMボード PIO-6BE1-A
定価 ¥25,000



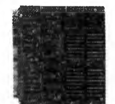
特価¥19,500

●2MB増設RAMボード PIO-6BE2-2M
定価 ¥50,000



特価¥38,500

●4MB増設RAMボード PIO-6BE4-4M
定価 ¥88,000



特価¥67,000

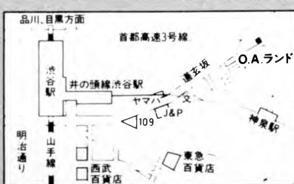
■ハードディスク ■特価品もありますのでTEL下さい。

●アイテック ITX-640	特価¥117,000	●シャープ CZ-620H	特価¥118,000
●アイテック ITX-680	特価¥149,000	●シャープ CZ-64H	特価¥ 95,000
●ロジテック LHD-32V	特価¥ 85,000	●アイテム HXD-040	特価¥ 88,000
●ロジテック LHD-34VE	特価¥ 90,000	●アイテム HXD-042	特価¥ 95,000
●ロジテック LHD-34V	特価¥104,000	●ICM SR-80	特価¥130,000

中古パソコン (価格/在庫は変動します。予約は5日以内とします。)

PC-9801RA5	¥338,000より	PC-286VS	¥165,000より
PC-9801RA2	¥265,000より	CZ-600C	¥160,000より
PC-9801RX2	¥199,000より	CZ-601C	¥170,000より
PC-9801EX2	¥190,000より	CZ-611C	¥198,000より
PC-9801VX21	¥170,000より	CZ-652C	¥178,000より
PC-9801UX21	¥165,000より	CZ-612C	¥210,000より
PC-9801VX2	¥160,000より	68000用モニター	¥ 49,000より
PC-9801VM21	¥150,000より	PC-9801用サウンドボード	¥ 13,000より
PC-9801UV11	¥148,000より	PC-88SR,FR	¥ 50,000より
PC-9801LV22	¥160,000より	PC-88FH,FA	¥ 65,000より
PC-286VE	¥150,000より	400ラインCRT	¥ 38,000より
PC-286US	¥155,000より	200ラインCRT	¥ 10,000より

通信販売のご案内
―全国通販―



■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。
〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド
■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1〜60回払で月々5,000円よりご自由に設定できます。

●下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
●ご注文、お問合せは… 午前10時から午後7時まで
●商品のお届けは…入金確認後、即日発送致します。

㈱オー・エー・ランド
〒150 東京都渋谷区円山町20-4 第5日新ビル1F
☎(03)770-8855 FAX (03)770-7080
関東エリアの送料は、1個につき¥1,000です。
★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合せ下さい。掲載の価格は、6月末現在です。

株式会社 デンキヤ



営業時間AM11:00〜PM7:00 水・木曜定休

（価格は全べて税込みです）

セット超特価

X68000
PERSONAL WORKSTATION

PROⅡ・PROⅡ HD

CZ-653C特価

CZ-663C特価

SUPER HD

CZ-623C特価

CZ-613D特価

セット超特価

X68000
PERSONAL WORKSTATION

EXPERTⅡ・EXPERTⅡ HD

CZ-603C特価

CZ-613C特価

EXPERT PRO

CZ-662C特価

CZ-602C特価

全品メーカー保証 即決クレジットOK

ディスプレイ		プリンタ		周辺機器		ソフト	
CZ-604D	特価	CZ-8PC4	特価	CZ-8NJ1	¥1,400	CZ-213MS	¥15,500
CZ-605D	特価	CZ-8PG1	特価	CZ-8NJ2	¥18,540	CZ-223CS	¥15,300
CZ-613D	特価	CZ-8PG2	特価	PIO-6BE1A	¥20,000	CZ-219SS	¥23,100
CU-21HD	特価	IO-735X	特価	PIO-6BE2	¥39,000	CZ-211LS	¥30,800

24時間テレホンサービス
0482-54-3444

お申し込み
TEL.0482-54-3400
FAX.0482-54-3443
埼玉県川口市西川口4-6-4

お支払い
下記取引銀行口座
までお振込み下さい。
三菱銀行西川口支店
㈱デンキヤ㊢0258081



- ●全商品完全保証書付(メーカー保証)
- ●全国無料配達(一部離島の方は有料になります)
- ●配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- ●どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- ●中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- ●お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(定休日▶渋谷店:日曜・祭日/横浜店:水曜)
AM10:00〜PM7:00

当社はX68000の販売認定店です。どんなことでも安心してご相談ください。

X68000 ビッグ・サマーセール開催中! 即売・即納

X68000 NEW PRO II

- ●CZ-653C(本体)……¥285,000
- ●CZ-603D(カラーディスプレイ)……¥ 84,800
- ●お好きなゲームソフト1本……¥ 7,800

■定価合計……¥377,600

クリエイト特価

均等払い	¥ 7,680×48回	¥ 9,890×36回	¥14,370×24回
ボーナス	なし	なし	なし

X68000 NEW EXPERT II

- ●CZ-603C(本体)……¥338,000
- ●CZ-613D(カラーディスプレイテレビ)……¥135,000
- ●CZ-8NJ2……¥ 23,800
- ●お好きなゲームソフト1本……¥ 9,800

■定価合計……¥506,600

クリエイト特価

均等払い	¥ 9,970×48回	¥12,840×36回	¥18,660×24回
ボーナス	なし	なし	なし

X68000 EXPERT II HD

- ●CZ-613C(本体)……¥448,000
- ●CZ-604D(カラーディスプレイ)……¥ 94,800
- ●お好きなゲームソフト1本……¥ 9,800

■定価合計……¥552,600

クリエイト特価

均等払い	¥ 5,920×48回	¥ 7,400×36回	¥12,100×24回
ボーナス	¥30,000× 8回	¥40,000× 6回	¥50,000× 4回

X68000 SUPER HD

- ●CZ-623C-TN(本体・キーボード・マウス)……¥498,000
- ●CZ-613D-TN(カラーディスプレイ)……¥135,000
- ●CZ-6BP1……¥ 79,800

■定価合計……¥712,800

クリエイト特価

均等払い	¥ 7,320×48回	¥10,100×36回	¥13,450×24回
ボーナス	¥42,000× 8回	¥50,000× 6回	¥80,000× 4回

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000 NEW EXPERT II

ミュージシャンセット。これもTMネットワークだよ〜!

- ●CZ-603C……¥338,000
- ●CZ-605D……¥115,000
- ●MU1.B(MIDIボード&ソフト)……¥ 39,800
- ●CM32L……¥ 69,000
- ●グラナダ……¥ 8,800
- ●JOYカード……¥ 1,800

■定価合計……¥572,400▶超特価¥458,000

X68000 NEW PRO II

ゲーマーズセット。遊んで暮らせるSET!

- ●PRO II CZ653C……¥285,000
- ●0.31CRT CZ603D……¥ 84,800
- ●グラナダ……¥ 8,800
- ●Y'S……¥ 8,700
- ●ポピュラス……¥ 9,800
- ●スーパーハングオン……¥ 8,800
- ●エージャックス……¥ 8,800
- ●サーク……¥ 8,800
- ●アールタイプ……¥ 7,800
- ●アナログJOYSTIC XE-1AP……¥ 13,800

■定価合計……¥445,100▶超特価¥353,000

X68000シリーズ用 周辺機器・ソフト オール超特価!!

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥ 69,800	MUSIC PRO	MIDI版	¥ 28,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥ 18,800
CZ-6BE1A	IMB増設RAMボード	¥ 38,000	SOUND PRO-68K	サウンドエディタ	¥ 15,800
CZ-6BE2	2MB増設RAMボード	¥ 79,800	Sampling PRO-68K	AD PCMサンプリングエディタ	¥ 17,800
CZ-6BE4	4MB増設RAMボード	¥138,000	Musicstudio PRO-68K V.1.1	MIDIマルチレコーディングソフト	¥ 28,800
CZ-8NM3	マウス・トラックボール	¥ 9,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥ 29,800
BF-68PRO	高性能CRTフィルター	¥ 19,800	PRO-68K	サイバーノート	¥ 19,800
CZ-6BP1	数値演算プロセッサ・ボード	¥ 79,800	PRO-68K	ステーショナリー	¥ 14,800
CZ-8NT1	トラックボール	¥ 13,800	Ccompiler PRO-68K	ソフト開発セット	¥ 39,800
CZ-6BM1	MIDIボード	¥ 26,800	Human 68K Ver2.0	開発ツールセット	¥ 9,800
CZ-8NJ2	アナログスティック	¥ 23,800	PIO-6BE1-A	内蔵1MRAM	¥ 25,000
CZ-6TU	パソコンチューナ	¥ 33,100	PIO-6BE2-2M	2MRAM	¥ 50,000
SX-68M	MIDI I/F	¥ 19,800	PIO-6BE4-4M	4MRAM	¥ 88,000
XE-1AP	アナログジョイパッド	¥ 13,800	MU1-B	MIDI I/F+ソフト	¥ 39,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。 ●超特価販売中!

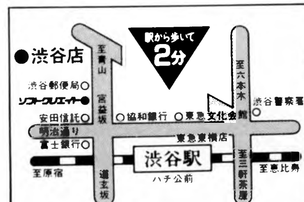
★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。

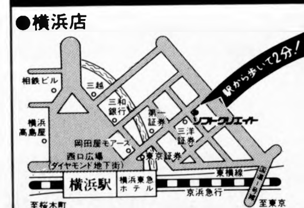
オール15%〜20%OFF

総合お問合せ先☎03-486-6541(代)

パソコン専門ショップ

ソフトクリエイト 渋谷/横浜





●渋谷店☎03-486-6541(代) 〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル 振込銀行:三井銀行 渋谷宮益坂支店(普)No.5000340
●横浜店☎045-314-4777(代) 〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル 振込銀行:三和銀行 横浜駅前支店(普)No.310852

X68000 全機種取り揃え大特価セール

AIBIT
アイビット電子株式会社

X68000 EXPERT/PRO



CZ-602C（本体）
＋CZ-603D（ディスプレイ）
＋SX.WINDOW
大特価¥310,000
（このセットに限り、送料＋消費税込）

CZ-653C（本体）
＋CZ-602D（ディスプレイ）
大特価¥288,000
（このセットに限り、送料＋消費税込）

※代金は商品引換着払いでもOKです。

New X68000新発売！（●特価価格は直接お問合せください。）

CZ-603C	定価¥338,000	〈ディスプレイ〉	
CZ-613C	定価¥448,000	CZ-603D	定価¥ 84,800
CZ-623C	定価¥498,000	CZ-604D	定価¥ 94,800
CZ-653C	定価¥285,000	CZ-605D	定価¥115,000
CZ-663C	定価¥395,000	CZ-613D	定価¥135,000

●新製品も大特価／お問合せください。

SHARPラップトップパソコン
AX286L-F



定価¥428,000⇨
特価¥238,000

FMTOWNSお買い得セット



1. TOWNS-1（本体）		¥338.000
2. FMT-ME(1M)（増設メモリー）	¥	60.000
3. FMD-FD301（増設FDユニット）	¥	28.000
4. FMT-KB101（キーボード）	¥	20.000
5. FMT-DP531（カラーディスプレイ）	¥	89.800
6. TOWNS-OS V1.1 L20	¥	20.000
定価合計		¥555.800

大特価！¥285,000

MZ2500下取り／MZ2500からMZ2861（定価¥328,000）に買い替え下取後 **特価¥165,000**
CZ600C下取り／CZ600CからCZ623（X68000 SUPER）に買い替え下取後 **特価¥300,000**

ハガキもOK、New MZプリンタ
漢字カラー熱転写プリンタ **シャープMZ-1P22**
好評発売中！



〈24×24ドット漢字●7色カラー●漢字30字／秒高速印字●MZ1P17とフルコンパチ●5KBのバッハメモリ付〉適応パソコン：MZ2000、2500、5500、6500シリーズ、X1シリーズ、X68000シリーズ他

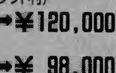
標準価格¥59,800➡ **特価¥25,000**

パソコンファクスMZ-1V01
"プリンタ・コピー・ファクス"
1台3役のスグレモノ
限定セット販売！



●MZ25セット（インターフェースソフト付）
標準価格合計¥342,800➡¥120,000
●MZ-1V01（本体のみ）
標準価格合計¥278,000➡¥ 98,000

シャープMZ-1X30 モデムホン
（1×19上位機種）



〈300/1200BPS 全2重通信対応モデム内蔵●音声入出力端子付●ダイヤルパルス／フッシュボタン対応●フッシュボタン音解析機能●シャーフ手順、CCITT、V25bis通信手順サポート〉

標準価格¥98,000➡ **大特価**

パソコンと専用ワープロをひとつにした16ビット
シャープMZ-2861



ワープロソフト『書院 28』
MS-DOS V3.1 装備
エミュレーションソフト搭載

定価¥328,000➡
大特価¥198,000
（ディスプレイは別）

MZ-2861用ソフト（UPシリーズ）
●1P-1251（デスクアップ）……定価¥88,000⇒特価¥20,000
●1P-1253（クリッパー）……定価¥77,000⇒特価¥20,000
●1P-1254（プランナップ）……定価¥88,000⇒特価¥20,000

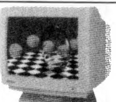
シャープMZ-2520
定価¥159,800➡**大特価¥78,000**

《在庫限り》
PC-E500PJ
定価¥28,800➡大特価
SHARPポケコン全機種、Z80ボード他、太平洋工業製品全機種取扱
●PC-E500PJご購入の方に、もれなく「ポケコンジャーナル特別号」を進呈！

PC-500と各種パソコンをつなぐインターフェースケーブル
CE-140T ¥8,800

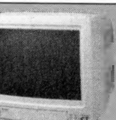
アイビット推奨ディスプレイ

●三菱XC-1498CII
（14型アナログ）
ドットピッチ0.28
定価¥107,000⇨
特価¥59,800



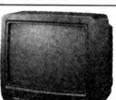
XC-1498CII対応パソコン機種：PC-9801シリーズ／PC-286シリーズ／PC-386シリーズ／PC-8801シリーズ
（上記機種には付属の接続ケーブルで、接続可能）

●シャープCZ-830D・BK
（14型）
2モードオートスキャン方式
（アナログ/デジタル）
定価¥98,000⇨
特価¥54,800



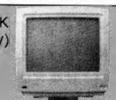
CZ-830D対応パソコン機種：CZ880C/881C、X1/TURBOシリーズ。ケーブルは本体付属を使用。NEC PC-8801・9801シリーズ（XA・XLのみ不可）MZ700/1500/2000/2200/2500各シリーズ（推奨品シャープ8D8K）。

●シャープCZ-602D-BK
（15型アナログTV/3モードオートスキャン）
定価¥99,800⇨
特価¥75,000



CZ-602D対応パソコン機種：※X1シリーズ／※X1 turboシリーズ／X1 yurboZシリーズ／X68000シリーズ／PC8801シリーズ／PC-9801シリーズ／PC-286シリーズ
（※は接続ケーブルAN1506が必要です）

●シャープ CZ-603D-GY・BK
（15型カラーディスプレイTV）
ドットピッチ3.9
定価¥84,800⇨
特価



CZ-603D対応パソコン機種：※X1シリーズ／※X1 turboシリーズ／X1 yurboZシリーズ／X68000シリーズ／PC8801シリーズ／PC-9801シリーズ／PC-286シリーズ
（※は接続ケーブルAN1506が必要です）

拡張機器他
●シャープCZ-8GR（X1 GRAM）……¥32,000⇨¥12,000
●シャープCZ-8EB3（I/Oボックス）……¥33,800⇨¥28,000
●シャープCZ-88K3（X1）……¥13,800⇨¥11,700
●シャープCZ-88K4（X1）……¥6,800⇨¥5,700
●シャープCZ-8BGR2（X1）……¥14,800⇨¥4,000
●シャープCZ-64H（ハードディスク）（CZ-602・652内蔵用）……特価
●シャープCZ-8NJ2（インテリジェントコントローラー）¥23,800⇨¥18,500
●シャープMZ-1E11……¥38,000⇨¥25,000
●シャープCZ-8ITチルトスタンド……¥8,500⇨¥1,000
●シャープMZ-1U08……¥25,000⇨¥12,000
●シャープMZ-1U03……¥35,000⇨¥15,000
●シャープMZ-1X22モデムユニット……¥21,800⇨¥13,000
●シャープMZ-1R12RAM……¥35,000⇨¥8,000
●シャープMZ-1E29（MZ）……¥17,800⇨¥9,800
●シャープMZ-1E30（2500ハードディスクインターフェイス）……¥25,000⇨¥22,500
●シャープMZ-1U09（2500）……¥9,000⇨¥7,200
●シャープMZ-1M03（5500）……¥69,000⇨¥35,000
●シャープMZ-8BC04（2000）……¥18,000⇨¥8,000
●シャープMZ-8B104（2000）……¥45,000⇨¥18,000
●シャープMZ-1R11（5500）……¥80,000⇨¥30,000
●シャープMZ-1R24（1500）……¥22,000⇨¥6,000
●シャープMZ-1R26A（2500）……¥13,000⇨¥12,800
●シャープMZ-1R27A（2500）……¥13,000⇨¥10,000
●シャープMZ-1R28A（2500）……¥13,000⇨¥10,000
●シャープMZ-1R29A（2500）……¥32,000⇨¥10,000
●シャープMZ-1T02（2200）……¥19,800⇨¥8,500
●シャープMZ-1T03（1500）……¥12,000⇨¥8,500
●シャープMZ-1X29……¥13,800⇨¥11,000
●テレシステムRM-25E（2500RAMファイル）……¥428,800⇨¥38,500
●シャープCZ-6ST1チルト台代品……特価¥3,500
●シャープCZ-8BE2（X1 320K RAM）……¥29,800⇨¥25,300
●シャープCZ-8BM2（X1 232Cマウスボード）……¥19,800⇨¥16,800
●シャープX1、MZ用マウス……特価¥4,800
●シャープX1用ジョイカード……¥1,500
●シャープMZ-5500キーボード……¥8,000
●シャープ2000/2200キーボード……¥8,000
●シャープMZ-1E08……¥9,000⇨¥8,000
●シャープCZ6BM1（X68用MIDIボード）¥26,800⇨¥23,000
●シャープCZ-6BE1（X68000増設RAM）……¥35,000⇨¥29,500
●シャープCZ-6BE1-A（X68000増設RAM）……¥38,000⇨¥23,800
●アイティックPIO-6BE1A（IOデータメモリー拡張ボード）……¥25,000⇨¥21,500
●アイティックPIO-6BE2-2M（IOデータメモリー拡張ボード）¥50,000⇨¥42,500
●アイティックPIO-6BE4-4M（IOデータメモリー拡張ボード）¥88,000⇨¥74,500

（MZ-2861）
●シャープMZ1R35（1MB増設メモリーボード）……¥55,000⇨¥19,000
●シャープMZ1R36（MZ-2861増設1MRAM）……¥45,000⇨¥15,000
●シャープSS-SC28M（ハンディーコピーキット）……¥49,800⇨¥10,000

プリンター
●シャープCZ-8PC4（黒・グレー）……¥99,800⇨大特価
●シャープCZ-8PG1……¥130,000⇨¥100,000
●シャープCZ-8PG2……¥160,000⇨¥130,000
●シャープMZ-1P27……¥268,000⇨¥214,400
●シャープMZ-1P28……¥148,000⇨¥118,400
●シャープMZ-1P29……¥168,000⇨¥134,400
●シャープMZ-6P18……¥60,000⇨¥35,000
●シャープMZ-6P27……¥58,000⇨¥39,800
●シャープMZ-6P29……¥50,000⇨¥37,500

フロッピーディスク
●シャープCZ-501H（X68拡張用10M）¥258,000⇨¥60,000
●シャープCZ-503F……¥49,800⇨¥30,000
●シャープCZ-53F……¥19,800⇨¥9,800
●シャープCZ-300F（CZ-3PCM付）……¥13,000

ハードディスク
●アイテックIT-X640……¥158,000⇨¥128,000
●アイテックIT-X68……¥198,000⇨¥158,000

ディスプレイ
●シャープMZ-1D17（15型カラーディスプレー）……¥124,000⇨¥63,000
●シャープMZ-1D27……¥120,000⇨¥79,800

ソフト
（X68000用）
●CZ-230ASニュージーランド……¥8,800⇨¥7,040
●CZ-230AS FULL. THRTTLE……¥8,800⇨¥7,040
●CZ-233AS PACMANIA……¥7,800⇨¥6,250
●CZ-222AS ARKANOID……¥7,800⇨¥6,250
●POPULOUS……¥9,800⇨¥7,850
●CZ-239AS THUNDARBLADE……¥9,500⇨¥8,000
●CZ-259SS X68000XWindow……特価

（MZ-2500用）
●1P-1215 COBOL……¥13,800⇨¥11,700
●DANGER BOX……¥5,800⇨¥2,000
●EXTRA HYPER DISK MONITOR……¥10,000⇨¥8,500
●EXTRA HYPER DISK MONITOR1……¥14,000⇨¥12,000
●FILE UTILITY〈UT-25F〉……¥6,800⇨¥6,000
●FREE CALL……¥6,800⇨¥1,000
●G-EDIT2500……¥8,000⇨¥7,000
●H.S.コントローラー……¥9,600⇨¥8,500
●HuCAL日本語……¥45,000⇨¥15,000
●カレイドスコープ……¥9,800⇨¥3,000
●カレイドスコープ2……¥5,800⇨¥1,000
●ザ・ブラックオニキス……¥7,800⇨¥3,000
●スーパー修理屋さん……¥12,000⇨¥10,200
●ムーンチャイルド……¥7,800⇨¥3,000
●英雄伝説サーガ……¥9,800⇨¥2,000
●五目並べ……¥4,800⇨¥2,000
●探検隊第2弾……¥7,800⇨¥2,000
●プリントSHOP……¥9,800⇨¥8,500
●プリントSHOPライブラリー1……¥4,500⇨¥3,800
●プリントSHOPライブラリー2……¥4,500⇨¥3,800

（X1用）
●日本語ワープロ将軍X1t……¥34,800⇨¥29,000
●日本語ワープロ侍X1t……¥19,800⇨¥16,800
●CZ-8WB51 X1tディスクBASIC……¥9,800⇨¥3,500
●3CP/M X1 3"CPM……¥16,800⇨¥5,000
●CZ-8BK3 X1t第二水準ROM……¥13,800⇨¥11,700
●CZ-128SF X1.CP/M……¥13,800⇨¥11,700
●CZ-130SF X1t CP/M……¥14,800⇨¥12,500
●CZ-116LF X1.C……¥13,800⇨¥11,700
●CZ-117S X1t LOGO……¥18,800⇨¥13,200
●CZ-118LF X1.COBOL……¥13,800⇨¥11,700
●CZ-126LF X1 APL……¥13,800⇨¥11,700
●CZ-115LF X1 FORTRAN……¥11,700

（MZ-5500、6500SOFT）
●MZ-2Z013（MZ-5500MSDOS）
●MZ-2Z014（MZ-5500TODAY）
●MZ-2Z023（MZ-5500GW.BASIC）
●MZ-2Z028（MZ-6500GW.BASIC）
●MZ-2Z025（MZ-5500ワープロ）
●MZ-2Z029（MZ-6500TODAY）

本体●シャープCZ-820、822、880、881、MZ-3500、2520、2861、2200、X68000、CZ-612、662、602、652 ●富士通FM-77AV-1、77AV-2、77AV-20、77AV-40●NEC PC-9801N ●東芝J3100SS

SHARPフリートップパソコン
All in Note 新発売！（入荷）
AX286N-H2 定価¥398,000

〈全商品新品完全保証付〉■シャープポケコン全商品販売中。カタログ、特価表ご請求ください（〒72）

☎0426-45-3001〜3
FAX.0426-44-6002
●営業時間/10:00〜19:00●電話受付/20:00迄可●定休日/日曜日（祭日営業）

SHARP SUPER XEX SHOP

アイビット電子株式会社 〒192 東京都八王子市北野町560-5

上記の広告商品はすべて店頭販売もしております。

全国通信販売
北海道 沖縄まで
す。
主文の際
于振込て
ております。
★お申し込みの際は必ず電話番号を明記して下さい。
富士
05

●本誌発売時には上記価格よりさらにお求めやすい価格に変更されている場合があります。●一部を除き、上記商品価格には消費税は含まれておりません。その商品に対し別途3%の消費税がかかりますのでご了承ください。

PERSONAL COMPUTER MAGAZINE
PC-9801活用誌
Oh!PC

パソコン●テスト●マガジン
PC MAGAZINE
JAPANESE EDITION

THE COMPUTER
コンピュータ時代を読む トレンド・マガジン

C MAGAZINE

ソフトバンクの雑誌が勢揃い

月刊 コンピュータ技術者必携 第2種・第1種・特種受験
情報処理試験

Oh!FM

Oh!X

BEEP! POWERFUL MEGA-MAGAZINE
MEGADRIVE
メガドライブ

《広告の半ページ》 ああ、また暑い夏がくる。♪ヘナヘナヘーナーヘェナ〜 ちからがぬーけーるー

月刊 電脳倶楽部 90年8月号(Vol.27) 7月18日発送

2HDディスクに入ったX68000のための雑誌だっ!

もしかして
CW送受信練習プログラム

そして
しりとりPRO-68K

さらに もしかして
X-BASIC配列サーチ・ソート関数

とどめはPDDで
軽犯罪法・日米安保条約

さらには
CZ-8PC2用
カラーハードコピープログラム

その他、便利なツール、PDD、ビープ音、読み物などを満載!
(なお、内容は一部変更されることがあります。ご了承下さい)

編集長祝一平からの御挨拶「どーもどーも、暑いですねぇ。今日なんかアイスキャンデーを五本も食べてしまいました。五本といえば龍角散」

満開製作所 電脳倶楽部 編集部
〒171 東京都豊島区要町1-19-3 いさみビル4F
TEL.(03)554-9282/FAX.(03)554-3856

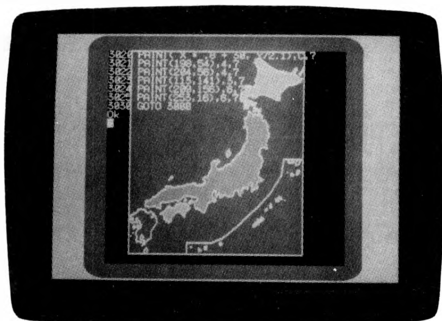
販売方法は通信販売のみです。お申し込みの方法は左記の住所へ現金書留で
定期購読 6ヶ月分 6,000円(消費税込・郵送料サービス)
●7月18日以降に受け付けた分は，原則としてVol.27から発送します。新たに購読を希望される方は，「新規」と御明記下さい。
●郵便振替を御利用の場合は口座番号「東京5-362847 満開製作所」でお願いいたします。
製品の性格上，返品には応じられませんが，お申し出があれば定期購読を解約し残金をお返しします。
(ご注意：バックナンバーの受け付けは，定期購読の方に限らせていただきます)

ACCESS

X1エミュレータ

好評発売中

定価¥9,800



X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3〜5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

X1エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。
  この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5" 2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したもの。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。
  このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

X1ディスク⟷X68000 Human68k(5" 2Dディスクイメージファイル)

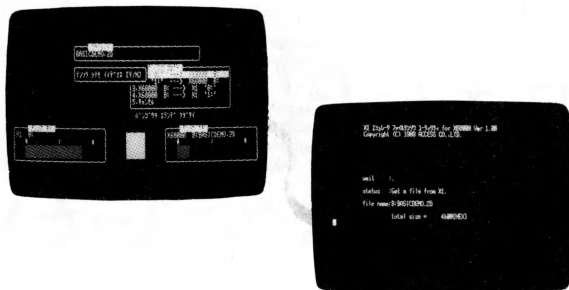
- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC : CP/M⟷X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。

※付属の専用ケーブルをX1とX68000に接続してファイルを転送します。



X1エミュレータ Q&A

Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?
A. 専用のケーブルが付属しますのでその必要はありません。

Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?
A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。

Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?
A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。

Q. Turbo用のソフトは動きますか?
A. X1用のみでTurbo専用のソフトは動きません。

Q. ゲームは動きますか?
A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。

＊タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
＊一部サポートしていない機能があります。

X1エミュレータ通信販売 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

＊この商品価格には消費税は含まれておりません。
＊CP/Mはデジタルリサーチ社の商標です。
　文中のソフトウェアは各社の商標です。
＊製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社アクセス 〒101 東京都千代田区神田神保町1-64 神保町協和ビル7F
☎03(233)0200(代) FAX.03(291)7019

資料請求券 oh!X 8月号

KAKUNETへようこそ!
KAKU! NET
WRITER'S PORT・KAKU! NET
SCENE1990.March.15 /Special Interest Group for WRITER'S

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

ライターズポート/書くネット!(ジャンプコード:WRITERS)



SIGのサブオペ(サブオペレーター)





ただいま実験的に
通信メディアで畑仕事の
機関紙づくりをしています

「書きこむ」人はみなライター!
プラスワンをめざす異業種交流SIG。

東京・大阪をまたにかけ、MSG(メッセージ)が飛び交ってゆく。ここは電脳ライター御用達、書き屋の港**「ライターズポート/書くネット!」**プランナーやらコピーライター・シナリオライターが、自由気ままに好き勝手な活動をしているから、**活動内容は多種多彩。各種広告・映像・出版物の批評/感想**なぞ朝メシ前で、電脳ライター必須の技術(?!)、ブラインドタッチなら、**自作のタイピング練習ソフトを開発する**という行動力! その他にも**自作の芝居を公演**したり、なぜだか**畑仕事(!)**にまで手を出してしまいます。**テーマはつねにプラスワン。「書きこむ」人はみなライター**と定義して、**異業種交流を基本**にパソコン通信と現実の活動とリンクさせているのです。書くこと、演劇、畑仕事、広告etc。興味のある方はこぞっておいでください。

その他 楽しいメニューがまだまだいっぱい!

- ★J&Pならではのパソコン・家電製品の会員割引もある**ONLINE SHOPPING**。
- ★J&Pだから強い!!パソコン情報をはじめとする役に立つ**DATA BASE**。
- ★みんなでおしゃべり**オンライントーク**(CHAT機能)。
- ★地域別・テーマ別ボードで充実の**BBS**(電子掲示板)。
- ★ビジュアルデータもばっちり送受信できる**X-MODEM**。

J&P HOT LINEへのご入会はスタータキットで。

買ったその日から
2週間無料で
アクセスできます。

お求めは、下記のお店へ。又は現金書留にて、¥3,000+¥90(消費税3%)=¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは 〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOT LINE事務局宛　TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

渋谷店	東京都渋谷区道玄坂2丁目28番4号	☎(03) 496-4141
町田店	東京都町田市森野1丁目39番16号	☎(0427)23-1313
八王子店	東京都八王子市旭町1番1号八王子そごう7F	☎(0426)26-4141
立川店	東京都立川市幸町4-39-1	☎(0425)36-4141
本厚木店	厚木市中町3-4-3	☎(0462)25-1548
富山店	富山市桜町2-1-10	☎(0764)32-3133
金沢店	金沢市入江2-63	☎(0762)91-1130
寺地店	金沢市寺地2-3	☎(0762)47-2524
大須店	名古屋市中区大須4丁目2-48	☎(052)262-1141
テクノランド	大阪市浪速区日本橋5丁目6番7号	☎(06) 634-1211
メディアランド	大阪市浪速区日本橋5丁目8番26号	☎(06) 634-1511
コスモランド	大阪市浪速区難波中2丁目1番17号	☎(06) 634-3111
U.S.LAND	大阪市浪速区日本橋4丁目9番15号	☎(06) 634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06) 348-1881
梅田店	大阪市北区小松原町1-10	☎(06) 362-1141
高槻店	高槻市高槻町11番16号	☎(0726)85-1212
くずは店	枚方市楠葉花園町15番2号	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3SENCHU PAL2番街4F	☎(06) 834-4141
摂津富田店	高槻市大畑町24-10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4-20	☎(0720)34-1166
藤井寺店	藤井寺市岡2丁目1番33号	☎(0729)38-2111
岸和田店	岸和田市土生町2451-3	☎(0724)37-1021
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
西宮店	兵庫県西宮市河原町5-11	☎(0798)71-1171
姫路店	姫路市東延末1丁目1番住友生命姫路南ビル1F	☎(0792)22-1221
京都寺町店	京都市下京区寺町通仏光寺下ル恵美須之町549	☎(075)341-3571
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
和歌山店	和歌山市元寺町4丁目4番地	☎(0734)28-1441
奈良1ばん館	奈良市三条町478-1	☎(0742)27-1111
郡山インター店	大和郡山市横田693-1	☎(07435)9-2221
熊本店	熊本市手取本町4-12	☎(096)359-7800

SHARP New-Life People

ADVANCED TURBO

先駆の“Z”アビリティがパソコンクリエイターを魅了する。



X1 パソコンテレビ turbo Z III

パーソナルコンピュータ＋キーボード＋マウス	CZ-888C-BK	標準価格	169,800円(税別)
14型カラーディスプレイテレビ	CZ-860D-BK	標準価格	92,200円(税別)
チルトスタンド	CZ-6ST1-B	標準価格	5,800円(税別)

クリエイティブマインドを刺激するAV機能 テレビ、ビデオ、ビデオディスクなどの映像を最大4,096色のリアルな画像で瞬時にグラフィック画面に取り込めるカラー画像デジタイズ機能を標準装備。4段階の量子化取り込み、42通りのモザイク取り込みなど多彩なトリック取り込み処理もサポート。さらにクロマキー合成、インターレーススーパーインポーズ、4,096色対応デジタルテロッパ機能、ステレオFM音源…先駆のAV機能がアートワークの領域をさらに拡げます。

AV指向の高水準ベーシックZ-BASIC搭載 多色グラフィック、カラー画像処理、ステレオFM音源、バンクメモリ対応など、ターボZシリーズが本来もつクリエイティブな機能をフルサポート。また豊富な画面モードで多色を駆使するときに便利なグラフィック用関数（HSV, RGB, HALF, CDOWN, CUP)も装備。さらにFM音源制御用ステートメントとしてX68000と命令コンパチの拡張MMLの採用によりスムーズな8音同時演奏を実現しています。

●メインメモリ128Kバイト標準装備、Z-BASICで最大576Kバイトまでサポート●1Mバイトの5インチフロッピーディスクドライブ2基搭載●JIS第1/第2水準準拠漢字、「システム・ユーザー辞書」を標準装備した高度な日本語処理機能●ニューデザインのマウス標準装備●X1ターボシリーズの豊富なソフト資産が活用できるコンパチブル設計●プリンタ、RS-232Cなど豊富なインターフェイスを装備●ドットピッチ0.39mmのハイコントラストブラウン管、15kHz/24kHzのデュアルスキャン方式採用14型カラーディスプレイテレビ(別売)。

資料請求券 X1turbo ZIII oh!X 8係

シャープ株式会社 ●お問い合わせは…シャープ(株)電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

T4910217908562 雑誌 02179-8

Printed in Japan